

In [1]:

```

import numpy as np
import dolfin as dl
from ufl import lhs, rhs, replace
import scipy.sparse as sps
import scipy.sparse.linalg as spla
import scipy.linalg as sla
import matplotlib.pyplot as plt
from pathlib import Path
from nalger_helper_functions import *
import meshio
import sys
from scipy.spatial import KDTree as scipy_KDTree
from scipy.optimize import root_scalar
from localpsf import localpsf_root
from localpsf.newtoncg import newtoncg_ls, cgsteihaug
from nalger_helper_functions import *

import hlibpro_python_wrapper as hpro
from localpsf.product_convolution_kernel import ProductConvolutionKernel
from localpsf.product_convolution_hmatrix import make_hmatrix_from_kernel
from localpsf.derivatives_at_point import StokesDerivativesAtPoint
from localpsf.bilaplacian_regularization import BiLaplacianRegularization

```

Options

In [2]:

```

# mesh_type = 'coarse'
# mesh_type = 'medium'
mesh_type = 'fine'
noise_level=1e-2
relative_prior_correlation_length=0.1
save_plots=True
gamma=1e4 # regularization gamma
m0_constant_value = 1.5 * 7.
rel_correlation_length = 0.1
reg_robin_bc=True
# mtrue_type = 'aces_building'
mtrue_type = 'angel_peak'
# mtrue_type = 'sinusoid'
# mtrue_string = 'm0 - (m0 / 7.)*std::cos((x[0]*x[0]+x[1]*x[1])*pi/(Radius*Radius)'
mtrue_string = 'm0 - (m0 / 7.)*std::cos(2.*x[0]*pi/Radius)'
solver_type = 'mumps'
# solver_type = 'default'
# solver_type = 'petsc'
# solver_type = 'umfpack'
# solver_type = 'superlu'
outflow_constant = 1.0e6

num_batches = 5 # number of batches used for Newton solves
tau = 3.0
num_neighbors = 10
hmatrix_tol = 1e-6
newton_rtol = 1e-8
# gamma_morozov = 15428.012899090285 # medium mesh, noise_level=1e-2
# gamma_morozov= 6932.714344119822 # fine mesh, noise_level=1e-2
# gamma_morozov= 6975.784285975133 # fine mesh, noise_level=1e-2
gamma_morozov = None

```

```
forcing_sequence_power = 0.5 # 1.0  
run_finite_difference_checks = False  
check_gauss_newton_hessian = False  
all_num_batches = [1, 5, 25] # number of batches used for spectral comparisons
```

In [3]:

```
save_dir = localpsf_root / 'numerical_examples' / 'stokes' / 'data'  
save_dir.mkdir(parents=True, exist_ok=True)  
save_dir_str = str(save_dir)
```

Mesh

Load unmodified mesh

In [4]:

```
if mesh_type == 'coarse':  
    mfile_name = str(localpsf_root) + "/numerical_examples/stokes/meshes/cylinder_coarse.xml"  
    lam = 1e10  
elif mesh_type == 'medium':  
    mfile_name = str(localpsf_root) + "/numerical_examples/stokes/meshes/cylinder_medium.xml"  
    lam = 1e11  
elif mesh_type == 'fine':  
    mfile_name = str(localpsf_root) + "/numerical_examples/stokes/meshes/cylinder_fine.xml"  
    lam = 1e12  
else:  
    raise RuntimeError('invalid mesh type '+mesh_type+', valid types are coarse, medium or fine')  
  
mesh = dl.Mesh(mfile_name+".xml")  
  
mesh
```

Out[4]:

Boundary submeshes and subdomains

In [5]:

```

 Options
 Help
 Summary
class BasalBoundary(dl.SubDomain):
    def inside(me, x, on_boundary):
        return dl.near(x[2], 0.) and on_boundary

class BasalBoundarySub(dl.SubDomain):
    def inside(me, x, on_boundary):
        return dl.near(x[2], 0.)

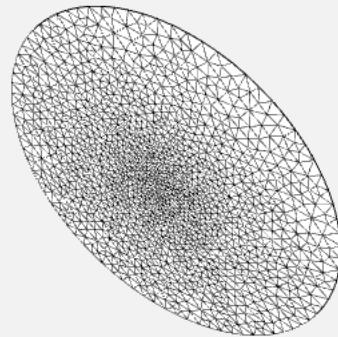
class TopBoundary(dl.SubDomain):
    def __init__(me, Height):
        me.Height = Height
        dl.SubDomain.__init__(me)
    def inside(me, x, on_boundary):
        return dl.near(x[1], me.Height) and on_boundary

boundary_markers = dl.MeshFunction("size_t", mesh, mfile_name+_facet_region.xml)
boundary_mesh = dl.BoundaryMesh(mesh, "exterior", True)
basal_mesh3D = dl.SubMesh(boundary_mesh, BasalBoundarySub())

basal_mesh3D

```

Out[5]:



Modify mesh

In [6]:

```

 Options
 Help
 Summary
r0 = 0.05
sig = 0.4
valleys = 4
valley_depth = 0.35
bump_height = 0.2

```

```
min_thickness = 0.08 / 8.
avg_thickness = 0.2 / 8.
theta = -np.pi / 2.
max_thickness = avg_thickness + (avg_thickness - min_thickness)
A_thickness = max_thickness - avg_thickness

dilitation = 1.e4
Length = 1.
Width = 1.
Length *= 2 * dilitation
Width *= 2 * dilitation
Radius = dilitation

prior_correlation_length = relative_prior_correlation_length * Radius

coords = mesh.coordinates()
bcoords = boundary_mesh.coordinates()
subbcoords = basal_mesh3D.coordinates()
coord_sets = [coords, bcoords, subbcoords]

def topography(r, t):
    zero = np.zeros(r.shape)
    R0 = r0 * np.ones(r.shape)
    return bump_height * np.exp(-(r / sig) ** 2) * (
        1. + valley_depth * np.sin(valleys * t - theta) * np.fmax(zero,

def depth(r, t):
    zero = np.zeros(r.shape)
    R0 = r0 * np.ones(r.shape)
    return min_thickness - A_thickness * np.sin(valleys * t - theta) * np.exp(
        -(r / sig) ** 2) * np.fmax(zero, (r - R0) / sig)

for k in range(len(coord_sets)):
    for i in range(len(coord_sets[k])):
        x, y, z = coord_sets[k][i]
        r = np.sqrt(x ** 2 + y ** 2)
        t = np.arctan2(y, x)
        coord_sets[k][i, 2] = depth(r, t) * z + topography(r, t)
        coord_sets[k][i] *= dilitation

mesh
```

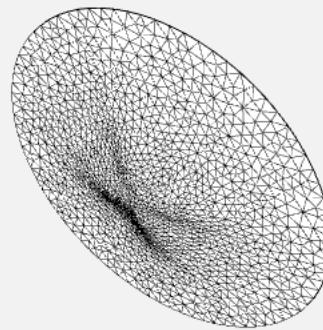
Out[6]:

In [7]:

```
Options  
basal_mesh3D
```

Summary

Out[7]:



2D Basal mesh

Summary

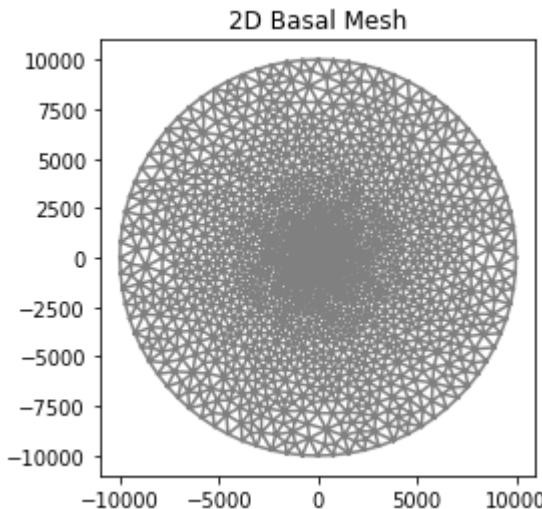
In [8]:

```
# ----- generate 2D mesh from 3D boundary subset mesh
coords = basal_mesh3D.coordinates()[:, :2]
cells = [("triangle", basal_mesh3D.cells())]
mesh2D = meshio.Mesh(coords, cells)
mesh2D.write("mesh2D.xml")
basal_mesh2D = dl.Mesh("mesh2D.xml")

dl.plot(basal_mesh2D)
plt.title('2D Basal Mesh')
```

WARNING:root:DOLFIN XML is a legacy format. Consider using XDMF instead.

Out[8]:



Plots to make (1):

- 3D mesh, colors, light mesh edges
- 2D mesh top-view, edges only, black and white

In [9]:

```
# Save mesh
dl.File(save_dir_str + "/ice_mountain_mesh3D.xml") << mesh
dl.File(save_dir_str + "/ice_mountain_mesh_base3D.xml") << basal_mesh3D
dl.File(save_dir_str + "/ice_mountain_mesh_base2D.xml") << basal_mesh2D
```

Function spaces

In [10]:

```
P1 = dl.FiniteElement("Lagrange", mesh.ufl_cell(), 1)
P2 = dl.VectorElement("Lagrange", mesh.ufl_cell(), 2)
TH = P2 * P1

Zh = dl.FunctionSpace(mesh, TH) # Zh: state, also adjoint
Wh = dl.FunctionSpace(mesh, 'Lagrange', 1) # Wh: parameter, full 3D do
Vh3 = dl.FunctionSpace(basal_mesh3D, 'Lagrange', 1) # Vh3: parameter, 2d basal n
Vh2 = dl.FunctionSpace(basal_mesh2D, 'Lagrange', 1) # Vh2: parameter, 2d basal n
Xh = [Zh, Wh, Zh] # Xh: (state, parameter full)
```

Transfer operators between function spaces

In [11]:

```
def function_space_prolongate_numpy(x_numpy, dim_Yh, inds_Xh_in_Yh):
    y_numpy = np.zeros(dim_Yh)
    y_numpy[inds_Xh_in_Yh] = x_numpy
    return y_numpy

def function_space_restrict_numpy(y_numpy, inds_Xh_in_Yh):
    return y_numpy[inds_Xh_in_Yh].copy()

def function_space_prolongate_petsc(x_petsc, Yh, inds_Xh_in_Yh):
    y_petsc = dl.Function(Yh).vector()
    y_petsc[:] = function_space_prolongate_numpy(x_petsc[:], Yh.dim(), inds_Xh_i
```

```

    return y_petsc

def function_space_restrict_petsc(y_petsc, Xh, inds_Xh_in_Yh):
    x_petsc = dl.Function(Xh).vector()
    x_petsc[:] = function_space_restrict_numpy(y_petsc[:], inds_Xh_in_Yh)
    return x_petsc

def make_prolongation_and_restriction_operators(Vsmall, Vbig, inds_Vsmall_in_Vbig,
    Vbig_to_Vsmall_numpy = lambda vb: function_space_restrict_numpy(vb, inds_Vsm)
    Vsmall_to_Vbig_numpy = lambda vs: function_space_prolongate_numpy(vs, Vbig)

    Vbig_to_Vsmall_petsc = lambda vb: function_space_restrict_petsc(vb, Vsmall,
    Vsmall_to_Vbig_petsc = lambda vs: function_space_prolongate_petsc(vs, Vbig,

    return Vbig_to_Vsmall_numpy, Vsmall_to_Vbig_numpy, Vbig_to_Vsmall_petsc, Vsm

```

In [12]:

```

pp_Vh3 = Vh3.tabulate_dof_coordinates()
pp_Vh2 = Vh2.tabulate_dof_coordinates()
pp_Wh = Wh.tabulate_dof_coordinates()

KDT_Wh = scipy_KDTree(pp_Wh)
inds_Vh3_in_Wh = KDT_Wh.query(pp_Vh3)[1]
if np.linalg.norm(pp_Vh3 - pp_Wh[inds_Vh3_in_Wh, :]) / np.linalg.norm(pp_Vh3) >
    warnings.warn('problem with basal function space inclusion')

pp_Vh3_2D = pp_Vh3[:, :2]

KDT_Vh3_2D = scipy_KDTree(pp_Vh3_2D)
inds_Vh2_in_Vh3 = KDT_Vh3_2D.query(pp_Vh2)[1]
if np.linalg.norm(pp_Vh2 - pp_Vh3[inds_Vh2_in_Vh3, :2]) / np.linalg.norm(pp_Vh2) >
    warnings.warn('inconsistency between manifold basal mesh and flat basal mesh')

inds_Vh2_in_Wh = inds_Vh3_in_Wh[inds_Vh2_in_Vh3]

Wh_to_Vh3_numpy, Vh3_to_Wh_numpy, Wh_to_Vh3_petsc, Vh3_to_Wh_petsc = \
    make_prolongation_and_restriction_operators(Vh3, Wh, inds_Vh3_in_Wh)

Vh3_to_Vh2_numpy, Vh2_to_Vh3_numpy, Vh3_to_Vh2_petsc, Vh2_to_Vh3_petsc = \
    make_prolongation_and_restriction_operators(Vh2, Vh3, inds_Vh2_in_Vh3)

Wh_to_Vh2_numpy, Vh2_to_Wh_numpy, Wh_to_Vh2_petsc, Vh2_to_Wh_petsc = \
    make_prolongation_and_restriction_operators(Vh2, Wh, inds_Vh2_in_Wh)

```

Forward Stokes forms

In [13]:

```

m_Wh = dl.Function(Wh) # Basal sliding friction
u = dl.Function(Zh) # Stokes state: (velocity, pressure)
p = dl.Function(Zh) # Stokes adjoint variable

mtrue_Wh = dl.Function(Wh) # True parameter
utru = dl.Function(Zh) # True state
uobs = dl.Function(Zh) # observed state (true state plus noise)

grav = 9.81           # acceleration due to gravity
rho = 910.0           # volumetric mass density of ice
stokes_forcing = dl.Constant( (0., 0., -rho*grav) )

```

```

ds = dl.Measure("ds", domain=mesh, subdomain_data=boundary_markers)
ds = ds
ds_base = ds(1)
ds_top = ds(2)
ds_lateral = ds(3)
normal = dl.FacetNormal(mesh)
# Strongly enforced Dirichlet conditions. The no outflow condition will be enforced
bcs = []
bcs0 = []

# Define the Nonlinear Stokes varfs
# rheology
stokes_n = 1.0
stokes_A = dl.Constant(2.140373e-7) # dl.Constant(1.e-16)
smooth_strain = dl.Constant(1e-6)

velocity, pressure = dl.split(u)
strain_rate = dl.sym(dl.grad(velocity))
normEul2 = 0.5 * dl.inner(strain_rate, strain_rate) + smooth_strain

tangent_velocity = (velocity - dl.outer(normal, normal)*velocity)

energy_t1 = stokes_A ** (-1.) * normEul2 * dl.dx
energy_t2 = -dl.inner(stokes_forcing, velocity) * dl.dx
energy_t3 = dl.Constant(.5) * dl.inner(dl.exp(m_Wh) * tangent_velocity, tangent_velocity)
energy_t4 = lam * dl.inner(velocity, normal) ** 2 * ds_base

energy_t5 = dl.Constant(outflow_constant) * dl.inner(velocity, normal)**2 * ds_top

energy = energy_t1 + energy_t2 + energy_t3 + energy_t4 + energy_t5
energy_gradient = dl.derivative(energy, u, p)

adjoint_velocity, adjoint_pressure = dl.split(p)
div_constraint = dl.inner(-dl.div(velocity), adjoint_pressure)
div_constraint_transpose = dl.inner(-dl.div(adjoint_velocity), pressure)

forward_form_ff = energy_gradient + div_constraint + div_constraint_transpose #

```

Misfit forms

In [14]:

```

def Tang(vel, n):
    return vel - dl.outer(n, n)*vel

dummy_u1 = dl.Function(Zh)
dummy_u2 = dl.Function(Zh)

dummy_velocity1, dummy_pressure1 = dl.split(dummy_u1)
dummy_velocity2, dummy_pressure2 = dl.split(dummy_u2)

data_inner_product_form = dl.inner(Tang(dummy_velocity1, normal),
                                   Tang(dummy_velocity2, normal))*ds_top

def data_inner_product(u_petsc, v_petsc):
    dummy_u1.vector()[:] = u_petsc[:]
    dummy_u2.vector()[:] = v_petsc[:]
    return dl.assemble(data_inner_product_form)

```

```

true_velocity, true_pressure = dl.split(utru)
observed_velocity, observed_pressure = dl.split(uobs)

velocity_discrepancy = observed_velocity - velocity

misfit_form = 0.5*dl.inner(Tang(velocity_discrepancy, normal),
                           Tang(velocity_discrepancy, normal))*ds_top

```

Lagrangian, adjoint, gradient, hessian

In [15]:

```

def input_vector_transformation(m_Vh2_numpy):
    m_Vh2_petsc = dl.Function(Vh2).vector()
    m_Vh2_petsc[:] = m_Vh2_numpy
    return Vh2_to_Wh_petsc(m_Vh2_petsc)

def output_vector_transformation(m_Wh_petsc):
    m_Vh2_numpy = Wh_to_Vh2_petsc(m_Wh_petsc)[:]
    return m_Vh2_numpy

SDAP = StokesDerivativesAtPoint(misfit_form, forward_form_ff, bcs,
                                 m_Wh, u, p,
                                 input_vector_transformation,
                                 output_vector_transformation,
                                 solver_type=solver_type)

```

Solve for true observations

In [16]:

```

if mtrue_type == 'angel_peak' or mtrue_type == 'aces_building':
    image_dir = localpsf_root / 'localpsf'

    if mtrue_type == 'angel_peak':
        image_file = image_dir / 'angel_peak_badlands_cropped9.png' # good
        image_file = image_dir / 'angel_peak_badlands_cropped7.png'
    else:
        image_file = image_dir / 'aces_building.png'

    mtrue_Vh2 = load_image_into_fenics(Vh2, image_file)
    minc = 9
    maxc = 12
    mtrue_Vh2_vec = mtrue_Vh2.vector()[:]
    mtrue_Vh2_vec *= 3
    mtrue_Vh2_vec += 9
    mtrue_Vh2.vector()[:] = mtrue_Vh2_vec
elif mtrue_type == 'sinusoid':
    mtrue_expr = dl.Expression(mtrue_string, element=Vh2.ufl_element(), m0=m0_cc)
    mtrue_Vh2 = dl.interpolate(mtrue_expr, Vh2)
else:
    raise RuntimeError('BAD mtrue_type')

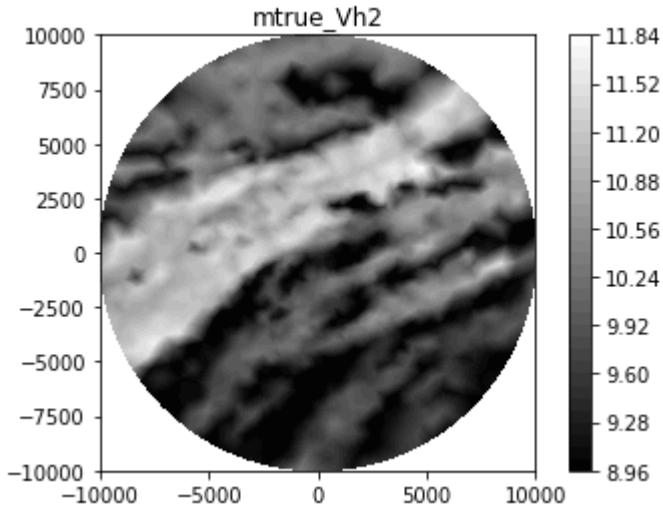
mtrue_Vh3 = dl.Function(Vh3)
mtrue_Vh3.vector()[:] = Vh2_to_Vh3_numpy(mtrue_Vh2.vector()[:])

cm = dl.plot(mtrue_Vh2, cmap='gray')

```

```
plt.colorbar(cm)
plt.title('mtrue_Vh2')
```

Out[16]: Text(0.5, 1.0, 'mtrue_Vh2')



In [17]:

```
SDAP.update_m(mtrue_Vh2.vector()[:])
SDAP.update_forward()
utruve.vector()[:] = SDAP.u.vector()[:].copy()
```

Plots to make (2):

- True velocity (arrows or not, up to you)
- mtrue (basal sliding friction field)

In [18]:

```
dl.File(save_dir_str + "/mtrue_Vh2.pvd") << mtrue_Vh2
dl.File(save_dir_str + "/mtrue_Vh3.pvd") << mtrue_Vh3

dl.File(save_dir_str + "/true_velocity" + str(outflow_constant)+".pvd") << utruve
dl.File(save_dir_str + "/true_pressure" + str(outflow_constant)+".pvd") << utruve
```

Add noise

In [19]:

```
utruve_vec = utruve.vector()[:]
noise_vec = noise_level * np.random.randn(Zh.dim()) * np.abs(utruve_vec)
uobs.vector()[:] = utruve_vec + noise_vec
```

In [20]:

```
def compute_misfit_datanorm():
    return np.sqrt(2.0 * SDAP.misfit())

noise_vec_petsc = dl.Function(Zh).vector()
noise_vec_petsc[:] = noise_vec
noise_datanorm = np.sqrt(data_inner_product(noise_vec_petsc, noise_vec_petsc))
print('noise_datanorm=', noise_datanorm)
```

noise_datanorm= 863.5355316705534

Regularization

In [21]:

```
m_prior_mean_Vh2 = dl.Function(Vh2)
m_prior_mean_Vh2.vector()[:] = m0_constant_value * np.ones(Vh2.dim())
m_Vh2 = dl.Function(Vh2)
m_Vh2.vector()[:] = Wh_to_Vh2_numpy(m_Wh.vector()[:])

REG = BiLaplacianRegularization(gamma, prior_correlation_length, m_Vh2, m_prior_mean_Vh2)

/home/nick/anaconda3/envs/fenics4/lib/python3.9/site-packages/scipy/sparse/linalg/dsolve/linsolve.py:318: SparseEfficiencyWarning: splu requires CSC matrix format
  warn('splu requires CSC matrix format', SparseEfficiencyWarning)
/home/nick/anaconda3/envs/fenics4/lib/python3.9/site-packages/scipy/sparse/coo.py:431: SparseEfficiencyWarning: Constructing a DIA matrix with 1416 diagonals is inefficient
  warn("Constructing a DIA matrix with %d diagonals "%
```

In [22]:

```
REG.update_gamma(1e5)
REG.update_correlation_length(prior_correlation_length)
```

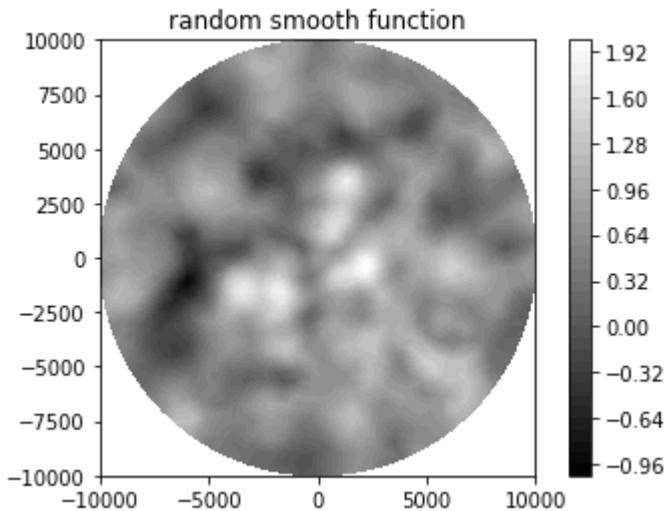
In [23]:

```
def random_smooth_Vh2_function(min_f, max_f):
    f_vec = REG.solve_R_numpy(np.random.randn(Vh2.dim()))
    f_vec = f_vec - np.min(f_vec)
    f_vec = (max_f - min_f) * f_vec / np.max(f_vec)
    f_vec = f_vec + min_f
    f = dl.Function(Vh2)
    f.vector()[:] = f_vec
    return f

random_smooth = random_smooth_Vh2_function(-1.0, 2.0)

cm = dl.plot(random_smooth, cmap='gray')
plt.colorbar(cm)
plt.title('random smooth function')
```

Out[23]:



In [24]:

```
def update_m(new_m_Vh2_numpy):
    SDAP.update_m(new_m_Vh2_numpy)
    REG.parameter.vector()[:] = new_m_Vh2_numpy

def cost():
    return SDAP.misfit() + REG.cost()

def gradient():
    return SDAP.gradient() + REG.gradient_numpy()

def apply_hessian(z_Vh2_numpy):
    return SDAP.apply_hessian(z_Vh2_numpy) + REG.apply_hessian_numpy(z_Vh2_numpy)

def apply_gauss_newton_hessian(z_Vh2_numpy):
    return SDAP.apply_gauss_newton_hessian(z_Vh2_numpy) + REG.apply_hessian_numpy(z_Vh2_numpy)
```

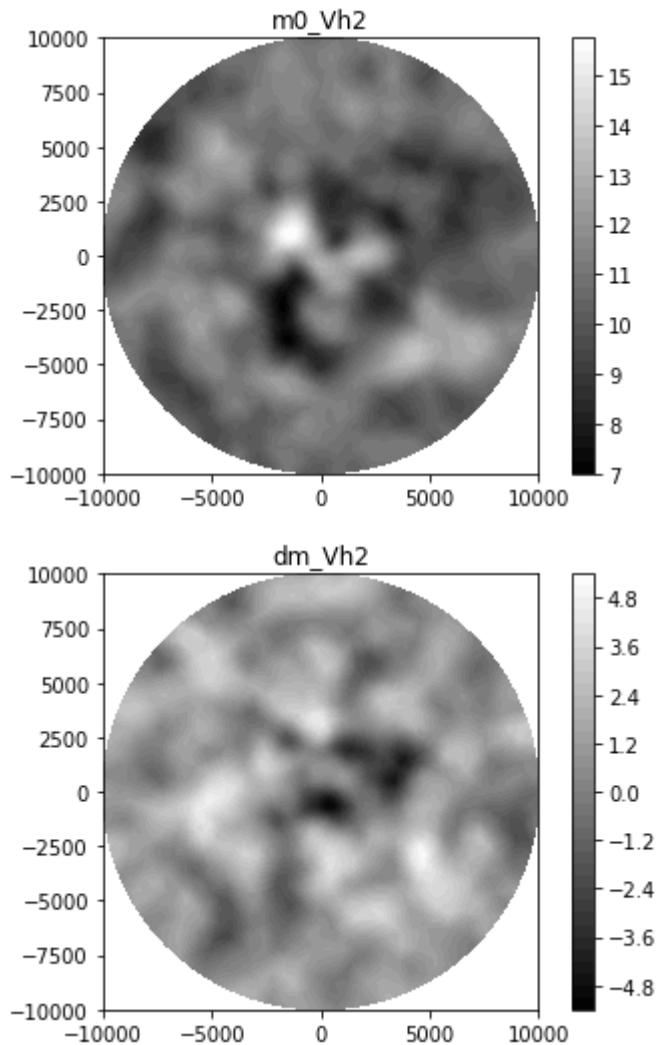
Finite difference check

In [25]:

```
m0_Vh2 = random_smooth_Vh2_function(m0_constant_value/1.5, m0_constant_value*1.5)
plt.figure()
cm = dl.plot(m0_Vh2, cmap='gray')
plt.colorbar(cm)
plt.title('m0_Vh2')

plt.figure()
dm_Vh2 = random_smooth_Vh2_function(-m0_constant_value/2.0, m0_constant_value/2.0)
cm = dl.plot(dm_Vh2, cmap='gray')
plt.colorbar(cm)
plt.title('dm_Vh2')
```

Out[25]:



In [26]:

```
if run_finite_difference_checks:
    m0_Vh2_numpy = m0_Vh2.vector()[:]
    dm_Vh2_numpy = dm_Vh2.vector()[:]

    update_m(m0_Vh2_numpy)
    J0_d = SDAP.misfit()
    g0_d = SDAP.gradient()
    H0dm_d = SDAP.apply_hessian(dm_Vh2_numpy)

    J0_r = REG.cost()
    g0_r = REG.gradient_numpy()
    H0dm_r = REG.apply_hessian_numpy(dm_Vh2_numpy)

    J0 = cost()
    g0 = gradient()
    H0dm = apply_hessian(dm_Vh2_numpy)

# ss = [1e-6]
ss = [1e0, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10, 1e-11]
errs_grad_d = []
errs_hess_d = []
errs_grad_r = []
errs_hess_r = []
errs_grad = []
errs_hess = []
for s in ss:
```

```

m1_Vh2_numpy = m0_Vh2_numpy + s * dm_Vh2_numpy
update_m(m1_Vh2_numpy)
J1_d = SDAP.misfit()
g1_d = SDAP.gradient()
#     H1dm_d = SDAP.apply_hessian(dm_Vh2_numpy)

J1_r = REG.cost()
g1_r = REG.gradient_numpy()
#     H1dm_r = REG.apply_hessian_numpy(dm_Vh2_numpy)

J1 = cost()
g1 = gradient()
#     H1dm = apply_hessian(dm_Vh2_numpy)

# Gradient

dJ_diff_d = (J1_d - J0_d) / s
dJ_d = np.dot(g0_d, dm_Vh2_numpy)
err_grad_d = np.abs(dJ_diff_d - dJ_d) / np.abs(dJ_diff_d)
errs_grad_d.append(err_grad_d)

dJ_diff_r = (J1_r - J0_r) / s
dJ_r = np.dot(g0_r, dm_Vh2_numpy)
err_grad_r = np.abs(dJ_diff_r - dJ_r) / np.abs(dJ_diff_r)
errs_grad_r.append(err_grad_r)

dJ_diff = (J1 - J0) / s
dJ = np.dot(g0, dm_Vh2_numpy)
err_grad = np.abs(dJ_diff - dJ) / np.abs(dJ_diff)
errs_grad.append(err_grad)

# Hessian

dg_diff_d = (g1_d - g0_d) / s
err_hess_d = np.linalg.norm(dg_diff_d - H0dm_d) / np.linalg.norm(dg_diff_d)
errs_hess_d.append(err_hess_d)

dg_diff_r = (g1_r - g0_r) / s
err_hess_r = np.linalg.norm(dg_diff_r - H0dm_r) / np.linalg.norm(dg_diff_r)
errs_hess_r.append(err_hess_r)

dg_diff = (g1 - g0) / s
err_hess = np.linalg.norm(dg_diff - H0dm) / np.linalg.norm(dg_diff)
errs_hess.append(err_hess)

print('s=' , s)
print('err_grad_d=' , err_grad_d, ', err_grad_r=' , err_grad_r, ', err_grad=' , err_grad)
print('err_hess_d=' , err_hess_d, ', err_hess_r=' , err_hess_r, ', err_hess=' , err_hess)

ss = np.array(ss)
errs_grad_d = np.array(errs_grad_d)
errs_hess_d = np.array(errs_hess_d)
errs_grad_r = np.array(errs_grad_r)
errs_hess_r = np.array(errs_hess_r)
errs_grad = np.array(errs_grad)
errs_hess = np.array(errs_hess)

plt.figure()
plt.loglog(ss, errs_grad_d)
plt.loglog(ss, errs_hess_d)
plt.title('finite difference check for misfit')

```

```

plt.xlabel('step size s')
plt.ylabel('error')
plt.legend(['misfit gradient', 'misfit hessian'])

plt.figure()
plt.loglog(ss, errs_grad_r)
plt.loglog(ss, errs_hess_r)
plt.title('finite difference check for regularization')
plt.xlabel('step size s')
plt.ylabel('error')
plt.legend(['regularization gradient', 'regularization hessian'])

plt.figure()
plt.loglog(ss, errs_grad)
plt.loglog(ss, errs_hess)
plt.title('finite difference check for overall cost')
plt.xlabel('step size s')
plt.ylabel('error')
plt.legend(['gradient', 'hessian'])

```

Check Gauss-Newton Hessian

In [27]:

```

if check_gauss_newton_hessian:
    m0_Vh2_numpy = m0_Vh2.vector()[:]

    uobs_vec_backup = uobs.vector()[:].copy()
    uobs.vector()[:] = utrue.vector()[:].copy()

    update_m(mtrue_Vh2.vector()[:].copy())

    z_Vh2_numpy = random_smooth_Vh2_function(-m0_constant_value/2.0, m0_constant)

    Hz = SDAP.apply_hessian(z_Vh2_numpy)
    HGNz = SDAP.apply_gauss_newton_hessian(z_Vh2_numpy)

    GN_discrepancy_at_truth_nonoise = np.linalg.norm(Hz - HGNz) / np.linalg.norm(Hz)
    print('GN_discrepancy_at_truth_nonoise=', GN_discrepancy_at_truth_nonoise)

    update_m(m0_Vh2_numpy)

    Hz = SDAP.apply_hessian(z_Vh2_numpy)
    HGNz = SDAP.apply_gauss_newton_hessian(z_Vh2_numpy)

    GN_discrepancy_elsewhere_nonoise = np.linalg.norm(Hz - HGNz) / np.linalg.norm(Hz)
    print('GN_discrepancy_elsewhere_nonoise=', GN_discrepancy_elsewhere_nonoise)

    uobs.vector()[:] = uobs_vec_backup

    update_m(mtrue_Vh2.vector()[:].copy())
    Hz = SDAP.apply_hessian(z_Vh2_numpy)
    HGNz = SDAP.apply_gauss_newton_hessian(z_Vh2_numpy)

    GN_discrepancy_at_truth_yesnoise = np.linalg.norm(Hz - HGNz) / np.linalg.norm(Hz)
    print('GN_discrepancy_at_truth_yesnoise=', GN_discrepancy_at_truth_yesnoise)

    update_m(m0_Vh2_numpy)
    Hz = SDAP.apply_hessian(z_Vh2_numpy)

```

```

HGNz = SDAP.apply_gauss_newton_hessian(z_Vh2_numpy)

GN_discrepancy_elsewhere_yesnoise = np.linalg.norm(Hz - HGNz) / np.linalg.norm(Hz)
print('GN_discrepancy_elsewhere_yesnoise=', GN_discrepancy_elsewhere_yesnoise)

# SDAP.update_m(m0_Vh2_numpy)
SDAP.update_forward()

p_Vh2_numpy = random_smooth_Vh2_function(-m0_constant_value/2.0, m0_constant_value)
q_Vh2_numpy = random_smooth_Vh2_function(-m0_constant_value/2.0, m0_constant_value)

SDAP.update_z(p_Vh2_numpy)
SDAP.update_incremental_forward()
du_dm_p = dl.Function(Zh)
du_dm_p.vector()[:] = SDAP.du_dm_z.vector()[:].copy()

SDAP.update_z(q_Vh2_numpy)
SDAP.update_incremental_forward()
du_dm_q = dl.Function(Zh)
du_dm_q.vector()[:] = SDAP.du_dm_z.vector()[:].copy()

C1 = data_inner_product(du_dm_p.vector(), du_dm_q.vector())

Hgn_p = SDAP.apply_gauss_newton_hessian(p_Vh2_numpy)
C2 = np.dot(q_Vh2_numpy, Hgn_p)

gauss_newton_error = np.abs(C1 - C2) / np.abs(C1)

print('gauss_newton_error=', gauss_newton_error)

```

In [28]:

```

def apply_misfit_gauss_newton_hessian_petsc(u_petsc):
    v = dl.Function(Vh2).vector()
    v[:] = SDAP.apply_gauss_newton_hessian(u_petsc[:])
    return v

class PCHWrapper:
    def __init__(me, do_updates=False):
        me.do_updates = do_updates

        me.PCK = None
        me.Hd_pch = None
        me.R_hmatrix = None
        me.H_pch = None
        me.preconditioner = None

    def build(me):
        me._make_Hd_hmatrix()
        me._make_reg_hmatrix()
        me._make_pre_hmatrix()

    def _make_pre_hmatrix(me):
        me.H_pch = me.Hd_pch + me.R_hmatrix
        me.preconditioner = me.H_pch.factorized_inverse()

    def _make_Hd_hmatrix(me):
        print('building PCH preconditioner')
        me.PCK = ProductConvolutionKernel(Vh2, Vh2,

```

```

apply_misfit_gauss_newton_hessian_pets
apply_misfit_gauss_newton_hessian_pets
num_batches, num_batches,
tau_rows=tau, tau_cols=tau,
num_neighbors_rows=num_neighbors,
num_neighbors_cols=num_neighbors)
Hd_pch_nonsym, extras = make_hmatrix_from_kernel(me.PCK, hmatrix_tol=hma
me.Hd_pch = Hd_pch_nonsym.spd()

def _make_reg_hmatrix(me):
    print('Building Regularization H-Matrix')
    me.R_hmatrix = REG.make_R_hmatrix(me.Hd_pch.bct, rtol=1e-6)

def update_reg(me):
    if me.Hd_pch is None:
        me._make_Hd_hmatrix()
    me._make_reg_hmatrix()
    me._make_pre_hmatrix()

def update_Hd(me):
    me._make_Hd_hmatrix()
    if me.R_hmatrix is None:
        me._make_reg_hmatrix()
    me._make_pre_hmatrix()

def low_rank_update(me, xx, yy):
    if me.do_updates and (me.preconditioner_hmatrix is not None):
        pass
        # me.preconditioner_hmatrix = me.preconditioner_hmatrix.

def solve(me, u_Vh2_numpy):
    if me.preconditioner is None:
        return REG.solve_hessian_numpy(u_Vh2_numpy)
    else:
        return me.preconditioner.matvec(u_Vh2_numpy)

def reset(me):
    me.PCK = None
    me.Hd_pch = None
    me.R_hmatrix = None
    me.H_pch = None
    me.preconditioner = None

```

In [29]:

```

REG.update_gamma(1e4)

get_optimization_variable = lambda : Wh_to_Vh2_numpy(m_Wh.vector()[:])
set_optimization_variable = update_m
cost_triple = lambda : [cost(), SDAP.misfit(), REG.cost()]
build_nothing = lambda : None
update_nothing = lambda X, Y: None

```

In [30]:

```

PCH1 = PCHWrapper(do_updates=False)

def morozov_helper():
    misfit_datanorm = np.sqrt(2.0 * SDAP.misfit())
    print('gamma=', REG.gamma, ', noise_datanorm=', noise_datanorm, ', misfit_da

```

```

    return misfit_datanorm

morozov_gammas = list()
morozov_discrepancies = list()

def log_morozov_function(log_gamma):
    gamma = np.exp(log_gamma)
    print()
    print('----- gamma=' , gamma, ' -----')
    REG.update_gamma(gamma)

    preconditioner_build_iters = (3,)
    if PCH1.preconditioner is None:
        preconditioner_build_iters = (3,)
    else:
        PCH1.update_reg()
        preconditioner_build_iters = tuple([])

    update_m(m0_constant_value * np.ones(Vh2.dim()))
    newtoncg_ls(get_optimization_variable,
                set_optimization_variable,
                cost_triple,
                gradient,
                apply_hessian,
                apply_gauss_newton_hessian,
                PCH1.build,
                update_nothing,
                PCH1.solve,
                preconditioner_build_iters = preconditioner_build_iters,
                rtol=newton_rtol,
                forcing_sequence_power=forcing_sequence_power)

    misfit_datanorm = morozov_helper()
    morozov_gammas.append(gamma)
    morozov_discrepancies.append(misfit_datanorm)

    log_mismatch = np.log(noise_datanorm) - np.log(misfit_datanorm)
    return log_mismatch

if gamma_morozov is None:
    sol_morozov = root_scalar(log_morozov_function, x0=np.log(5e3), x1=np.log(5e6))
    print(sol_morozov)
    gamma_morozov = np.exp(sol_morozov.root)
    print('gamma_morozov=' , gamma_morozov)

REG.update_gamma(gamma_morozov)

----- gamma= 5000.000000000004 -----
using Gauss-Newton Hessian
it= 0 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 2.528e+05
(M r0, r0) = 2.037e+06
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 60800772279799.06
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 7797484.9970871415

```

```
=====
Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha  t
olcg
 0   1   1   1   1   T   F  2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07  ---- 5.0
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.906e+06
||M r0|| = 5.409e+04
(M r0, r0) = 5.845e+05
Iteration : 0 (r, r) = 62500719723087.62
Iteration : 1 (r, r) = 25868298506353.766
Iteration : 2 (r, r) = 15319779174433.328
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 3914048.948906149

=====
Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
```

```

Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0    : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1    : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07      ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01

converged : False
reason      : unknown reason
cumulative CG iterations : 3
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 3
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.876e+06
||M r0|| = 2.271e+04
(M r0, r0) = 2.391e+05
Iteration : 0 (r, r) = 15021920577626.777
Iteration : 1 (r, r) = 9845330889144.463
Iteration : 2 (r, r) = 7737089442423.863
Iteration : 3 (r, r) = 4611240518703.695
Iteration : 4 (r, r) = 3497989937372.867
Iteration : 5 (r, r) = 2796040040126.543
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 1672136.3700746847

===== Begin Newton CG convergence information =====

```

```
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise
e.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07    ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.178416e+07 2.2e+07 2.6e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 8
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 8
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
building preconditioner
building PCH preconditioner
Computing impulse response moments
getting spatially varying volume
getting spatially varying mean
getting spatially varying covariance
Preparing sample point batch stuff
Preparing c++ object
Building initial sample point batches
```

/home/nick/repos/localpsf/localpsf/sample_point_batches.py:94: NumbaPerformanceW

```

arning: np.dot() is faster on contiguous arrays, called on (array(float64, 2d,
A), array(float64, 2d, C))
  if ellipsoids_intersect(Sigma_perm[x_ind, :, :], Sigma_perm[p_ind, :, :], mu_p
erm[x_ind, :, :],
Making row and column cluster treesUsing ProductConvolutionKernelRBFColsOnly!
Making block cluster trees

Building A kernel hmatrix
— building H-matrix ( tol = Making input and output mass matrix hmatrices
Computing A_hmatrix = M_out_hmatrix * A_kernel_hmatrix * M_in_hmatrix
making hmatrix spdle-06 )
      ==-]    7% ETA 11 s (5.12 GB)
symmetrizing                                         getting smallest
eigenvalue with Lanczos
  done in 12.97s
  size of H-matrix = 4.42 MB
lambda_min= -76580.83327169948
scaling_at_lambda_min= 0.9878048780487805
scaling_at_zero= 0.5
Setting up operator T = (2*A - (b+a) I) / (b-a)
computing T^(2^k)
computing T^(2^1) = T^(2^0) * T^(2^0)

computing T^(2^2) = T^(2^1) * T^(2^1)
computing negative spectral projector Pi_minus = I / (I + T^(2^k))— H-matrix mu
ltiplication C=A*B
  done in 33.53s
  size of C = 5.27 MB

— H-matrix multiplication C=A*B
  done in 34.77s
  size of C = 7.70 MB

— H-matrix inverse ( rtol = 1e-07 , atol = 1e-12 , overwrite= False )
                                         97397613525
  size of inverse = 14640401 bytes

computing A_plus = Pi * A

— H-matrix multiplication C=A*B
  done in 37.30s
  size of C = 17.53 MB
Building Regularization H-Matrix
  matrix has dimension 1416 x 1416
  no of non-zeroes      = 9746
  matrix is              real valued
  format                 = non symmetric
  size of sparse matrix = 163.52 kB
  |S|_F                  = 806144
  sparsity constant = 20
  size of H-matrix     = 4.06 MB
  |A|_F                  = 806144
  |S-A|_2 = 5.4232e-09
  matrix has dimension — H-matrix inverse ( rtol = 1e-06 , atol = 1e-14 , ove
rwrite= False )
  1416 x 1416
  no of non-zeroes      = 9746
  matrix is              real valued
  format                 = symmetric
  size of sparse matrix = 163.52 kB

```

```

|S|_F          = 5.18561e+06
  sparsity constant = 20
  size of H-matrix = 2.29 MB
|A|_F          = 5.18561e+06
|S-A|_2 = 4.66653e-09
[==                                         done in 1
5.418750524520874
  size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
  done in 24.67s
] 4% ETA 27 s (5.20 GB)
  size of C = 3.86 MB

— H-matrix multiplication C=A*B
  done in 33.28s
  size of C = 8.53 MB

Residual: r=b-Ax
— LU factorisation ( rtol = 1e-07 )
[==--                                         Preconditioner: M =approx= A^-1
||r0|| = 1.627e+06
||M r0|| = 3.933e+01
(M r0, r0) = 3.905e+03
  Iteration : 0 (r, r) = 2646144442184.8745
[===== (r, r) = 366414301859.5593
[========== 2 (r, r) = 57802220637.35317
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 240420.9238759247
[===== ] 89% ETA 3 s (5.21 GB)
===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t

```

```

olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07      ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.178416e+07 2.2e+07 2.6e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3   2   1   1   2   T   T 6.165938e+06 6.1e+06 6.1e+04 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 10
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 10
===== End Newton CG convergence information =====
=====

done in 31.24s
size of LU factor = 18.69 MB
using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.617e+05
||M r0|| = 1.288e+01
(M r0, r0) = 1.027e+03
Iteration : 0 (r, r) = 68483623043.65376
Iteration : 1 (r, r) = 31981559678.06558
Iteration : 2 (r, r) = 10761652533.239012
Iteration : 3 (r, r) = 3809862148.703622
Iteration : 4 (r, r) = 1112038312.778342
Iteration : 5 (r, r) = 571265944.5628774
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 23901.17035968903

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd

```

```

reg:      regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg  (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07  ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.178416e+07 2.2e+07 2.6e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3   2   1   1   2   T   T 6.165938e+06 6.1e+06 6.1e+04 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
  4   5   1   1   5   T   F 8.218143e+05 5.4e+05 2.8e+05 -5.1e+05 2.6e+05 1.0e+00 1.2
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 15
cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 15
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.487e+04
||M r0|| = 1.403e+00
(M r0, r0) = 8.386e+01
Iteration : 0 (r, r) = 1215958116.5525784
Iteration : 1 (r, r) = 417884304.99576837
Iteration : 2 (r, r) = 128318492.13023134
Iteration : 3 (r, r) = 151795373.92534086
Iteration : 4 (r, r) = 104411095.17273203
Iteration : 5 (r, r) = 30524202.764465027
Iteration : 6 (r, r) = 8843132.773494504
Iteration : 7 (r, r) = 5275252.058082572
Iteration : 8 (r, r) = 3148887.3993596765
Iteration : 9 (r, r) = 1756242.544646705
Relative/Absolute residual less than tol
Converged in 9 iterations with final norm 1325.233015226645

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...

```

```

it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07   ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.178416e+07 2.2e+07 2.6e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3   2   1   1   2   T   T 6.165938e+06 6.1e+06 6.1e+04 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
  4   5   1   1   5   T   F 8.218143e+05 5.4e+05 2.8e+05 -5.1e+05 2.6e+05 1.0e+00 1.2
e-01
  5   9   1   1   9   T   F 5.696227e+05 3.3e+05 2.4e+05 -7.1e+03 3.5e+04 1.0e+00 4.2
e-02

converged : False
reason      : unknown reason
cumulative CG iterations : 24
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 24
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.832e+03
||M r0|| = 9.013e-02
(M r0, r0) = 4.920e+00
Iteration : 0 (r, r) = 3356716.5020248247
Iteration : 1 (r, r) = 1446937.6878977364
Iteration : 2 (r, r) = 420612.97650490917
Iteration : 3 (r, r) = 321700.54030757665
Iteration : 4 (r, r) = 277535.755474856
Iteration : 5 (r, r) = 148023.123180567
Iteration : 6 (r, r) = 58260.53671776735
Iteration : 7 (r, r) = 29460.732394096784
Iteration : 8 (r, r) = 24558.39352351096
Iteration : 9 (r, r) = 18189.053385198655

```

```

Iteration : 10 (r, r) = 3128.9248763057767
Iteration : 11 (r, r) = 1285.0892650385576
Iteration : 12 (r, r) = 486.70348065888885
Iteration : 13 (r, r) = 212.35924359665927
Relative/Absolute residual less than tol
Converged in 13 iterations with final norm 14.572551032563217

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07   ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.178416e+07 2.2e+07 2.6e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3   2   1   1   2   T   T 6.165938e+06 6.1e+06 6.1e+04 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
  4   5   1   1   5   T   F 8.218143e+05 5.4e+05 2.8e+05 -5.1e+05 2.6e+05 1.0e+00 1.2
e-01
  5   9   1   1   9   T   F 5.696227e+05 3.3e+05 2.4e+05 -7.1e+03 3.5e+04 1.0e+00 4.2
e-02
  6  13   1   1  13   T   F 5.661010e+05 3.3e+05 2.4e+05 -2.5e+01 1.8e+03 1.0e+00 9.7
e-03

converged : False
reason    : unknown reason
cumulative CG iterations : 37
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 37
===== End Newton CG convergence information =====

```

```
=====
```

```
using Hessian
it= 7 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
False
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.749e+01
||M r0|| = 1.862e-03
(M r0, r0) = 7.587e-02
Iteration : 0 (r, r) = 305.8281629576305
Iteration : 1 (r, r) = 178.476651349541
Iteration : 2 (r, r) = 96.83062502098187
Iteration : 3 (r, r) = 52.936712304136634
Iteration : 4 (r, r) = 40.71405985354616
Iteration : 5 (r, r) = 19.06338122616416
Iteration : 6 (r, r) = 6.647026458057597
Iteration : 7 (r, r) = 2.078940355305485
Iteration : 8 (r, r) = 1.090015705544459
Iteration : 9 (r, r) = 0.5818700576444459
Iteration : 10 (r, r) = 0.2905726315017592
Iteration : 11 (r, r) = 0.1570359409728928
Iteration : 12 (r, r) = 0.07972613501548531
Iteration : 13 (r, r) = 0.04541808865839962
Iteration : 14 (r, r) = 0.01923156224586578
Iteration : 15 (r, r) = 0.00998209894438901
Iteration : 16 (r, r) = 0.004508057294932381
Iteration : 17 (r, r) = 0.001772705308016505
Iteration : 18 (r, r) = 0.0007139075328989931
Iteration : 19 (r, r) = 0.0003153627074175871
Iteration : 20 (r, r) = 0.00013280439322403394
Relative/Absolute residual less than tol
Converged in 20 iterations with final norm 0.011524078844924393
```

```
===== Begin Newton CG convergence information =====
```

```
=====
```

```
Preconditioned inexact Newton-CG with line search
```

```
Hp=-g
```

```
u <- u + alpha * p
```

```
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
```

```
it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

```
it:      Newton iteration number
```

```
nCG:     number of CG iterations in Newton iteration
```

```
nJ:      number of cost function evaluations in Newton iteration
```

```
nG:      number of gradient evaluations in Newton iteration
```

```
nHp:     number of Hessian-vector products in Newton iteration
```

```
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
```

```
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
```

```
cost:    cost, J = Jd + Jr
```

```
misfit:  misfit cost, Jd
```

```
reg:     regularization cost, Jr
```

```
(g,p):  inner product between gradient, g, and Newton search direction, p
```

```
||g||L2: l2 norm of gradient
```

```
alpha:   step size
```

tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	-----	5.0	
e-01	1	2	1	1	2	T	F	6.080028e+07	6.1e+07	1.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
e-01	2	5	1	1	5	T	F	2.178416e+07	2.2e+07	2.6e+04	-3.1e+07	3.9e+06	1.0e+00	4.5
e-01	3	2	1	1	2	T	T	6.165938e+06	6.1e+06	6.1e+04	-1.1e+07	1.6e+06	1.0e+00	2.9
e-01	4	5	1	1	5	T	F	8.218143e+05	5.4e+05	2.8e+05	-5.1e+05	2.6e+05	1.0e+00	1.2
e-02	5	9	1	1	9	T	F	5.696227e+05	3.3e+05	2.4e+05	-7.1e+03	3.5e+04	1.0e+00	4.2
e-03	6	13	1	1	13	T	F	5.661010e+05	3.3e+05	2.4e+05	-2.5e+01	1.8e+03	1.0e+00	9.7
e-04	7	20	1	1	20	F	F	5.660884e+05	3.3e+05	2.4e+05	-5.4e-03	1.7e+01	1.0e+00	9.5

converged : False

reason : unknown reason

cumulative CG iterations : 57

cumulative cost evaluations : 8

cumulative gradient evaluations : 8

cumulative Hessian vector products (excluding preconditioner builds) : 57

===== End Newton CG convergence information =====

=====

using Hessian

it= 8 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton= False

===== Begin Newton CG convergence information =====

=====

Preconditioned inexact Newton-CG with line search

$Hp=-g$

$u \leftarrow u + \alpha * p$

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : $u=u_0 \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $Hp=-g$

it=1 : linesearch $u \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $Hp=-g$

...

it=last : linesearch $u \rightarrow J \rightarrow g \rightarrow$ Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, $J = J_d + J_r$

misfit: misfit cost, J_d

reg: regularization cost, J_r

(g,p): inner product between gradient, g, and Newton search direction, p

$\|g\|L2$: l2 norm of gradient

```

alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost misfit      reg (g,p) ||g||L2 alpha t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07 ---- 5.0
e-01
  1   2   1   1   2   T   F 6.080028e+07 6.1e+07 1.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.178416e+07 2.2e+07 2.6e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3   2   1   1   2   T   T 6.165938e+06 6.1e+06 6.1e+04 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
  4   5   1   1   5   T   F 8.218143e+05 5.4e+05 2.8e+05 -5.1e+05 2.6e+05 1.0e+00 1.2
e-01
  5   9   1   1   9   T   F 5.696227e+05 3.3e+05 2.4e+05 -7.1e+03 3.5e+04 1.0e+00 4.2
e-02
  6  13   1   1  13   T   F 5.661010e+05 3.3e+05 2.4e+05 -2.5e+01 1.8e+03 1.0e+00 9.7
e-03
  7  20   1   1  20   F   F 5.660884e+05 3.3e+05 2.4e+05 -5.4e-03 1.7e+01 1.0e+00 9.5
e-04
  8   0   1   1   0   F   F 5.660884e+05 3.3e+05 2.4e+05 ---- 5.2e-02 1.0e+00
-----
converged : True
reason   : Norm of the gradient less than tolerance
cumulative CG iterations : 57
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 57
===== End Newton CG convergence information =====
=====

gamma= 5000.000000000004 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
812.6169880110655

----- gamma= 50000.00000000001 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format               = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                = 8.06144e+06
sparsity constant = 20
size of H-matrix   = 4.06 MB
|A|_F                = 8.06144e+06
|S-A|_2 = 6.76813e-09
matrix has dimension — H-matrix inverse ( rtol = 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format               = symmetric
size of sparse matrix = 163.52 kB
|S|_F                = 5.18561e+06
sparsity constant = 20
size of H-matrix   = 2.29 MB
|A|_F                = 5.18561e+06
|S-A|_2 = 4.66653e-09
1e-06 , atol = 1e-14 , overwrite= False )

```

```

/home/nick/anaconda3/envs/fenics4/lib/python3.9/site-packages/scipy/sparse/linal
g/dsolve/linsolve.py:318: SparseEfficiencyWarning: splu requires CSC matrix form
at
    warn('splu requires CSC matrix format', SparseEfficiencyWarning)
        done in 14.279305696487427
        size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
    done in 24.03s
    size of C = 3.95 MB

— H-matrix multiplication C=A*B
                                                done in 33.39s
    size of C = 8.64 MB

— LU factorisation ( rtol = 1e-07 )
[=====] n
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
[=====] Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 3.595e+01
(M r0, r0) = 2.169e+04
Iteration : 0 (r, r) = 380758959655851.75
    done in 30.63s
    size of LU factor = 17.94 MB
Iteration : 1 (r, r) = 22458528040128.277
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 4739042.945588094

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)

```

```

stokes_dev1
it nCG nJ nG nHp GN BP          cost  misfit      reg   (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.923e+06
||M r0|| = 1.007e+01
(M r0, r0) = 4.632e+03
Iteration : 0 (r, r) = 15392549306050.43
Iteration : 1 (r, r) = 352544684504.33435
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 593754.7343005649

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg   (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01

```

```
1 1 1 1 1 T F 2.047000e+07 1.4e+07 6.9e+06 -2.1e+07 3.9e+06 1.0e+00 4.5
e-01
```

```
converged : False
reason      : unknown reason
cumulative CG iterations : 2
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information =====
=====
```

```
using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 5.329e+05
||M r0|| = 3.243e+00
(M r0, r0) = 8.008e+02
Iteration : 0 (r, r) = 283941708606.2915
Iteration : 1 (r, r) = 17216423679.299385
Iteration : 2 (r, r) = 999462153.3520812
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 31614.271355703917
```

```
===== Begin Newton CG convergence information =====
=====
```

```
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion
```

```
it=0     : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1     : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.
```

```
it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)
```

it nCG nJ nG nHp GN BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0 1 1 1 1 T F 2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01							
1 1 1 1 1 T F 2.047000e+07	1.4e+07	6.9e+06	-2.1e+07	3.9e+06	1.0e+00	4.5	

```

e-01
 2 2 1 1 2 T F 1.002705e+07 3.5e+06 6.5e+06 -5.2e+05 5.3e+05 1.0e+00 1.7
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 4
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 4
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.363e+04
||M r0|| = 2.028e-01
(M r0, r0) = 5.799e+01
Iteration : 0 (r, r) = 1130766339.588983
Iteration : 1 (r, r) = 59144820.533889666
Iteration : 2 (r, r) = 7040928.623888069
Iteration : 3 (r, r) = 203541.26394750192
Relative/Absolute residual less than tol
Converged in 3 iterations with final norm 451.1554764684808

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha  t
olcg
 0  1 1 1 1 T F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07  ---- 5.0

```

```

e-01
  1  1  1  1   1  T F 2.047000e+07 1.4e+07 6.9e+06 -2.1e+07 3.9e+06 1.0e+00 4.5
e-01
  2  2  1  1   2  T F 1.002705e+07 3.5e+06 6.5e+06 -5.2e+05 5.3e+05 1.0e+00 1.7
e-01
  3  3  1  1   3  T F 9.770928e+06 3.3e+06 6.4e+06 -2.7e+03 3.4e+04 1.0e+00 4.2
e-02

converged : False
reason    : unknown reason
cumulative CG iterations : 7
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 7
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 9.132e+02
||M r0|| = 1.103e-02
(M r0, r0) = 2.542e+00
Iteration : 0 (r, r) = 833849.4027094174
Iteration : 1 (r, r) = 146794.71532708744
Iteration : 2 (r, r) = 9749.073084496502
Iteration : 3 (r, r) = 317.4747197346938
Iteration : 4 (r, r) = 17.518234131799126
Relative/Absolute residual less than tol
Converged in 4 iterations with final norm 4.185478960859692

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea

```

se)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t
0	1	1	1	1	T	F	$2.093839e+08$	$2.1e+08$	$0.0e+00$	$-3.6e+08$	$2.0e+07$	----	5.0
e-01	1	1	1	1	T	F	$2.047000e+07$	$1.4e+07$	$6.9e+06$	$-2.1e+07$	$3.9e+06$	$1.0e+00$	4.5
e-01	2	2	1	1	2	T	$1.002705e+07$	$3.5e+06$	$6.5e+06$	$-5.2e+05$	$5.3e+05$	$1.0e+00$	1.7
e-01	3	3	1	1	3	T	$9.770928e+06$	$3.3e+06$	$6.4e+06$	$-2.7e+03$	$3.4e+04$	$1.0e+00$	4.2
e-02	4	4	1	1	4	T	$9.769621e+06$	$3.3e+06$	$6.4e+06$	$-5.1e+00$	$9.1e+02$	$1.0e+00$	6.8
e-03													

converged : False

reason : unknown reason

cumulative CG iterations : 11

cumulative cost evaluations : 5

cumulative gradient evaluations : 5

cumulative Hessian vector products (excluding preconditioner builds) : 11

===== End Newton CG convergence information =====

=====

using Gauss-Newton Hessian

it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True

Residual: r=b-Ax

Preconditioner: M =approx= A^-1

 $\|r_0\| = 4.873e+01$ $\|M r_0\| = 7.853e-04$ $(M r_0, r_0) = 1.626e-01$

Iteration : 0 (r, r) = 2374.1513963100197

Iteration : 1 (r, r) = 594.3755554841518

Iteration : 2 (r, r) = 30.536713225260378

Iteration : 3 (r, r) = 0.9773040458769024

Iteration : 4 (r, r) = 0.04648430875090048

Iteration : 5 (r, r) = 0.003952444565034312

Relative/Absolute residual less than tol

Converged in 5 iterations with final norm 0.06286847035704235

===== Begin Newton CG convergence information =====

=====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g

it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g

...

it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise

```

e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 2.047000e+07 1.4e+07 6.9e+06 -2.1e+07 3.9e+06 1.0e+00 4.5
e-01
  2   2   1   1   2   T   F 1.002705e+07 3.5e+06 6.5e+06 -5.2e+05 5.3e+05 1.0e+00 1.7
e-01
  3   3   1   1   3   T   F 9.770928e+06 3.3e+06 6.4e+06 -2.7e+03 3.4e+04 1.0e+00 4.2
e-02
  4   4   1   1   4   T   F 9.769621e+06 3.3e+06 6.4e+06 -5.1e+00 9.1e+02 1.0e+00 6.8
e-03
  5   5   1   1   5   T   F 9.769619e+06 3.3e+06 6.4e+06 -2.0e-02 4.9e+01 1.0e+00 1.6
e-03

converged : False
reason   : unknown reason
cumulative CG iterations : 16
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 16
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.128e+00
||M r0|| = 5.626e-05
(M r0, r0) = 1.099e-02
  Iteration : 0 (r, r) = 9.785201657116076
  Iteration : 1 (r, r) = 3.008032134304214
  Iteration : 2 (r, r) = 0.12813413814545815
  Iteration : 3 (r, r) = 0.005516259996165365
  Iteration : 4 (r, r) = 0.0002661537357763828
  Iteration : 5 (r, r) = 1.9430069563418738e-05
  Iteration : 6 (r, r) = 1.3714377913856614e-06
Relative/Absolute residual less than tol
Converged in 6 iterations with final norm 0.0011710840240502223

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g

```

```

it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise
e.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07      ---- 5.0
e-01
  1   1   1   1   1   T   F 2.047000e+07 1.4e+07 6.9e+06 -2.1e+07 3.9e+06 1.0e+00 4.5
e-01
  2   2   1   1   2   T   F 1.002705e+07 3.5e+06 6.5e+06 -5.2e+05 5.3e+05 1.0e+00 1.7
e-01
  3   3   1   1   3   T   F 9.770928e+06 3.3e+06 6.4e+06 -2.7e+03 3.4e+04 1.0e+00 4.2
e-02
  4   4   1   1   4   T   F 9.769621e+06 3.3e+06 6.4e+06 -5.1e+00 9.1e+02 1.0e+00 6.8
e-03
  5   5   1   1   5   T   F 9.769619e+06 3.3e+06 6.4e+06 -2.0e-02 4.9e+01 1.0e+00 1.6
e-03
  6   6   1   1   6   T   F 9.769619e+06 3.3e+06 6.4e+06 -9.0e-05 3.1e+00 1.0e+00 4.0
e-04

converged : False
reason      : unknown reason
cumulative CG iterations : 22
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information =====
=====

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

```

```

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

```

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	-----	5.0
e-01													
1	1	1	1	1	T	F	2.047000e+07	1.4e+07	6.9e+06	-2.1e+07	3.9e+06	1.0e+00	4.5
e-01													
2	2	1	1	2	T	F	1.002705e+07	3.5e+06	6.5e+06	-5.2e+05	5.3e+05	1.0e+00	1.7
e-01													
3	3	1	1	3	T	F	9.770928e+06	3.3e+06	6.4e+06	-2.7e+03	3.4e+04	1.0e+00	4.2
e-02													
4	4	1	1	4	T	F	9.769621e+06	3.3e+06	6.4e+06	-5.1e+00	9.1e+02	1.0e+00	6.8
e-03													
5	5	1	1	5	T	F	9.769619e+06	3.3e+06	6.4e+06	-2.0e-02	4.9e+01	1.0e+00	1.6
e-03													
6	6	1	1	6	T	F	9.769619e+06	3.3e+06	6.4e+06	-9.0e-05	3.1e+00	1.0e+00	4.0
e-04													
7	0	10	1	0	T	F	9.769619e+06	3.3e+06	6.4e+06	-----	3.1e+00	9.8e-04	

```

converged : False
reason   : Maximum number of backtracking reached
cumulative CG iterations : 22
cumulative cost evaluations : 17
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information =====
=====
```

```
gamma= 50000.00000000001 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
2585.9744874650937
```

```

----- gamma= 5642.493956627751 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is              real valued
format                = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 909732
sparsity constant = 20
size of H-matrix  = 4.06 MB
|A|_F                 = 909732
|S-A|_2 = 2.77941e-09
matrix has dimension — H-matrix inverse ( rtol = 1416 x 1416
```

```

        no of non-zeroes      = 9746
        matrix is            real valued
        format                = symmetric
        size of sparse matrix = 163.52 kB
        |S|_F                 = 5.18561e+06
        sparsity constant = 20
        size of H-matrix   = 2.29 MB
        |A|_F                 = 5.18561e+06
        |S-A|_2 = 4.66653e-09
        1e-06 , atol = 1e-14 , overwrite= False  )
                                                0868759155

(5.21 GB)
    size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
    done in 24.08s
    size of C = 3.87 MB

— H-matrix multiplication C=A*B
    done in 32.96s
    size of C = 8.53 MB

— LU factorisation ( rtol = 1e-07 )
    done in 30.75s
    size of LU factor = 18.71 MB
using Gauss-Newton Hessian
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 6.356e+01
(M r0, r0) = 2.263e+04
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 27959776538156.125
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5287700.4962607445

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd

```

```

reg:      regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit  reg  (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.5e+08 2.0e+07  ---- 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 5.224e+06
||M r0|| = 6.381e+01
(M r0, r0) = 8.469e+03
Iteration : 0 (r, r) = 27289142408423.758
Iteration : 1 (r, r) = 7701060835297.035
Iteration : 2 (r, r) = 1311442969153.9053
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 1145182.5047362125

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient

```

```

alpha: step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost misfit reg (g,p) ||g||L2 alpha t
olcg
  0   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.5e+08 2.0e+07 ---- 5.0
e-01
  1   2   1   1   2   T   F 2.132606e+07 2.0e+07 1.5e+06 -3.7e+07 5.2e+06 1.0e+00 5.0
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 3
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 3
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.530e+06
||M r0|| = 3.820e+01
(M r0, r0) = 3.525e+03
Iteration : 0 (r, r) = 2340536424140.0137
Iteration : 1 (r, r) = 712914200013.5758
Iteration : 2 (r, r) = 175316183704.7832
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 418707.754531467

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size

```

tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.5e+08	2.0e+07	-----	5.0	
e-01	1	2	1	1	2	T	F	2.132606e+07	2.0e+07	1.5e+06	-3.7e+07	5.2e+06	1.0e+00	5.0
e-01	2	2	1	1	2	T	F	3.338753e+06	2.8e+06	4.9e+05	-4.3e+06	1.5e+06	1.0e+00	2.8
e-01														

converged : False

reason : unknown reason

cumulative CG iterations : 5

cumulative cost evaluations : 3

cumulative gradient evaluations : 3

cumulative Hessian vector products (excluding preconditioner builds) : 5

===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian

it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True

Residual: $r=b-Ax$

Preconditioner: $M = \text{approx} = A^{-1}$

$\|r_0\| = 4.986e+05$

$\|M r_0\| = 1.761e+01$

$(M r_0, r_0) = 1.161e+03$

Iteration : 0 $(r, r) = 248568164917.9255$

Iteration : 1 $(r, r) = 106842720955.19986$

Iteration : 2 $(r, r) = 20930111230.756157$

Iteration : 3 $(r, r) = 17866906665.744617$

Iteration : 4 $(r, r) = 14926724379.708942$

Iteration : 5 $(r, r) = 4978380900.0606365$

Relative/Absolute residual less than tol

Converged in 5 iterations with final norm 70557.64239301535

===== Begin Newton CG convergence information =====
=====

Preconditioned inexact Newton-CG with line search

$H_p = -g$

$u \leftarrow u + \alpha * p$

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : $u=u_0 \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $H_p = -g$

it=1 : linesearch $u \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $H_p = -g$

...

it=last : linesearch $u \rightarrow J \rightarrow g \rightarrow$ Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, $J = J_d + J_r$

misfit: misfit cost, J_d

```

reg:      regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg  (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.5e+08 2.0e+07   ---- 5.0
e-01
  1   2   1   1   2   T   F 2.132606e+07 2.0e+07 1.5e+06 -3.7e+07 5.2e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 3.338753e+06 2.8e+06 4.9e+05 -4.3e+06 1.5e+06 1.0e+00 2.8
e-01
  3   5   1   1   5   T   F 1.195733e+06 8.2e+05 3.7e+05 -1.0e+06 5.0e+05 1.0e+00 1.6
e-01

converged : False
reason   : unknown reason
cumulative CG iterations : 10
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 10
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 8.543e+04
||M r0|| = 5.409e+00
(M r0, r0) = 3.824e+02
  Iteration : 0 (r, r) = 7297845207.598768
  Iteration : 1 (r, r) = 2280685646.174207
  Iteration : 2 (r, r) = 659728017.0973521
  Iteration : 3 (r, r) = 225314734.4025255
  Iteration : 4 (r, r) = 89762691.02489269
  Iteration : 5 (r, r) = 57860846.27469475
  Iteration : 6 (r, r) = 48513613.62454468
  Iteration : 7 (r, r) = 21170138.144938618
Relative/Absolute residual less than tol
Converged in 7 iterations with final norm 4601.101840313754

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration

```

nJ: number of cost function evaluations in Newton iteration
 nG: number of gradient evaluations in Newton iteration
 nHp: number of Hessian-vector products in Newton iteration
 GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
 BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
 cost: cost, $J = J_d + J_r$
 misfit: misfit cost, J_d
 reg: regularization cost, J_r
 (g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|_{L2}$: l2 norm of gradient
 alpha: step size
 tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.5e+08	2.0e+07	----	5.0	
e-01	1	2	1	1	2	T	F	2.132606e+07	2.0e+07	1.5e+06	-3.7e+07	5.2e+06	1.0e+00	5.0
e-01	2	2	1	1	2	T	F	3.338753e+06	2.8e+06	4.9e+05	-4.3e+06	1.5e+06	1.0e+00	2.8
e-01	3	5	1	1	5	T	F	1.195733e+06	8.2e+05	3.7e+05	-1.0e+06	5.0e+05	1.0e+00	1.6
e-02	4	7	1	1	7	T	F	6.670863e+05	3.7e+05	3.0e+05	-7.6e+04	8.5e+04	1.0e+00	6.6

converged : False
 reason : unknown reason
 cumulative CG iterations : 17
 cumulative cost evaluations : 5
 cumulative gradient evaluations : 5
 cumulative Hessian vector products (excluding preconditioner builds) : 17
===== End Newton CG convergence information ======
=====

using Gauss-Newton Hessian
 it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
 Residual: r=b-Ax
 Preconditioner: M =approx= A^-1
 $\|r_0\| = 7.056e+03$
 $\|M r_0\| = 2.762e-01$
 $(M r_0, r_0) = 2.327e+01$
 Iteration : 0 (r, r) = 49792409.53401345
 Iteration : 1 (r, r) = 22010573.92844159
 Iteration : 2 (r, r) = 6471232.010882744
 Iteration : 3 (r, r) = 3532920.94038814
 Iteration : 4 (r, r) = 2141096.200956701
 Iteration : 5 (r, r) = 1358342.6692035976
 Iteration : 6 (r, r) = 778075.9056996037
 Iteration : 7 (r, r) = 281847.7694706188
 Iteration : 8 (r, r) = 100866.0569116978
 Iteration : 9 (r, r) = 30838.7852635023
 Iteration : 10 (r, r) = 9049.563496632909
 Relative/Absolute residual less than tol
 Converged in 10 iterations with final norm 95.1291937137749

===== Begin Newton CG convergence information ======
=====

```

Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.5e+08 2.0e+07   ---- 5.0
e-01
  1   2   1   1   2   T   F 2.132606e+07 2.0e+07 1.5e+06 -3.7e+07 5.2e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 3.338753e+06 2.8e+06 4.9e+05 -4.3e+06 1.5e+06 1.0e+00 2.8
e-01
  3   5   1   1   5   T   F 1.195733e+06 8.2e+05 3.7e+05 -1.0e+06 5.0e+05 1.0e+00 1.6
e-01
  4   7   1   1   7   T   F 6.670863e+05 3.7e+05 3.0e+05 -7.6e+04 8.5e+04 1.0e+00 6.6
e-02
  5  10   1   1  10   T   F 6.292693e+05 3.4e+05 2.9e+05 -4.5e+02 7.1e+03 1.0e+00 1.9
e-02

converged : False
reason     : unknown reason
cumulative CG iterations : 27
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 27
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.340e+02
||M r0|| = 6.322e-03
(M r0, r0) = 4.462e-01
Iteration : 0 (r, r) = 17958.22980205646

```

```

Iteration : 1 (r, r) = 6858.528240221176
Iteration : 2 (r, r) = 1876.9757529167366
Iteration : 3 (r, r) = 1065.5891962126195
Iteration : 4 (r, r) = 733.3525130555047
Iteration : 5 (r, r) = 344.57629869435925
Iteration : 6 (r, r) = 138.15005937735427
Iteration : 7 (r, r) = 76.76910479247397
Iteration : 8 (r, r) = 35.07656150156684
Iteration : 9 (r, r) = 14.76832484573847
Iteration : 10 (r, r) = 4.795889229333161
Iteration : 11 (r, r) = 1.9263074200520975
Iteration : 12 (r, r) = 0.8235434270191263
Iteration : 13 (r, r) = 0.4654588938497404
Iteration : 14 (r, r) = 0.254923188605036
Iteration : 15 (r, r) = 0.11286904328390368
Relative/Absolute residual less than tol
Converged in 15 iterations with final norm 0.3359598834442941

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.5e+08 2.0e+07    ---- 5.0
e-01
  1   2   1   1   2   T   F 2.132606e+07 2.0e+07 1.5e+06 -3.7e+07 5.2e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 3.338753e+06 2.8e+06 4.9e+05 -4.3e+06 1.5e+06 1.0e+00 2.8
e-01
  3   5   1   1   5   T   F 1.195733e+06 8.2e+05 3.7e+05 -1.0e+06 5.0e+05 1.0e+00 1.6
e-01
  4   7   1   1   7   T   F 6.670863e+05 3.7e+05 3.0e+05 -7.6e+04 8.5e+04 1.0e+00 6.6
e-02
  5   10  1   1  10   T   F 6.292693e+05 3.4e+05 2.9e+05 -4.5e+02 7.1e+03 1.0e+00 1.9

```

```

e-02
 6 15 1 1 15 T F 6.290439e+05 3.4e+05 2.9e+05 -1.6e-01 1.3e+02 1.0e+00 2.6
e-03

converged : False
reason    : unknown reason
cumulative CG iterations : 42
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 42
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.173e+00
||M r0|| = 2.675e-04
(M r0, r0) = 9.478e-03
  Iteration : 0 (r, r) = 1.3755405355551833
  Iteration : 1 (r, r) = 0.6227369434083823
  Iteration : 2 (r, r) = 0.22928756030759018
  Iteration : 3 (r, r) = 0.10367138361380868
  Iteration : 4 (r, r) = 0.06324886955755232
  Iteration : 5 (r, r) = 0.03711992955355433
  Iteration : 6 (r, r) = 0.018845951386799842
  Iteration : 7 (r, r) = 0.0077388123714759015
  Iteration : 8 (r, r) = 0.0036279920549297604
  Iteration : 9 (r, r) = 0.0013675002671183008
  Iteration : 10 (r, r) = 0.0005571189427543016
  Iteration : 11 (r, r) = 0.0002814548589007438
  Iteration : 12 (r, r) = 0.00013737385872023458
  Iteration : 13 (r, r) = 6.896216035485181e-05
  Iteration : 14 (r, r) = 3.024945284928301e-05
  Iteration : 15 (r, r) = 1.4963198146291305e-05
  Iteration : 16 (r, r) = 6.051372710972423e-06
  Iteration : 17 (r, r) = 2.7617118528609315e-06
  Iteration : 18 (r, r) = 1.4594959022473484e-06
  Iteration : 19 (r, r) = 7.161965143108112e-07
  Iteration : 20 (r, r) = 2.2493266982281057e-07
  Iteration : 21 (r, r) = 9.439702462987222e-08
  Iteration : 22 (r, r) = 4.069067152507951e-08
Relative/Absolute residual less than tol
Converged in 22 iterations with final norm 0.0002017192889266654

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number

```

nCG: number of CG iterations in Newton iteration
nJ: number of cost function evaluations in Newton iteration
nG: number of gradient evaluations in Newton iteration
nHp: number of Hessian-vector products in Newton iteration
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost: cost, $J = J_d + J_r$
misfit: misfit cost, J_d
reg: regularization cost, J_r
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.5e+08	2.0e+07	----	5.0
e-01	1	2	1	1	2	T	2.132606e+07	2.0e+07	1.5e+06	-3.7e+07	5.2e+06	1.0e+00	5.0
e-01	2	2	1	1	2	T	3.338753e+06	2.8e+06	4.9e+05	-4.3e+06	1.5e+06	1.0e+00	2.8
e-01	3	5	1	1	5	T	1.195733e+06	8.2e+05	3.7e+05	-1.0e+06	5.0e+05	1.0e+00	1.6
e-02	4	7	1	1	7	T	6.670863e+05	3.7e+05	3.0e+05	-7.6e+04	8.5e+04	1.0e+00	6.6
e-02	5	10	1	1	10	T	6.292693e+05	3.4e+05	2.9e+05	-4.5e+02	7.1e+03	1.0e+00	1.9
e-03	6	15	1	1	15	T	6.290439e+05	3.4e+05	2.9e+05	-1.6e-01	1.3e+02	1.0e+00	2.6
e-04	7	22	1	1	22	F	6.290438e+05	3.4e+05	2.9e+05	-5.1e-05	1.2e+00	1.0e+00	2.5

converged : False
reason : unknown reason
cumulative CG iterations : 64
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 64
===== End Newton CG convergence information ======

using Hessian
it= 8 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= False
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.029e+00
||M r0|| = 2.340e-04
(M r0, r0) = 8.291e-03
Iteration : 0 (r, r) = 1.0582034675173972
Iteration : 1 (r, r) = 0.47877975463098
Iteration : 2 (r, r) = 0.1758227178656292
Iteration : 3 (r, r) = 0.07950096551186883
Iteration : 4 (r, r) = 0.04860318715260439
Iteration : 5 (r, r) = 0.028465697646442217
Iteration : 6 (r, r) = 0.01445260068055998
Iteration : 7 (r, r) = 0.005926764608446447
Iteration : 8 (r, r) = 0.0027776855731647095

```

Iteration :  9  (r, r) =  0.0010475326373439976
Iteration : 10  (r, r) =  0.00042735154789226256
Iteration : 11  (r, r) =  0.00021617944665006133
Iteration : 12  (r, r) =  0.00010554465227443668
Iteration : 13  (r, r) =  5.296695753497628e-05
Iteration : 14  (r, r) =  2.3154808605604264e-05
Iteration : 15  (r, r) =  1.1445675389251117e-05
Iteration : 16  (r, r) =  4.639358406142161e-06
Iteration : 17  (r, r) =  2.119967894102724e-06
Iteration : 18  (r, r) =  1.1188220520013279e-06
Iteration : 19  (r, r) =  5.482384253147205e-07
Iteration : 20  (r, r) =  1.7232321902324986e-07
Iteration : 21  (r, r) =  7.238109400925625e-08
Iteration : 22  (r, r) =  3.111892231841758e-08

```

Relative/Absolute residual less than tol

Converged in 22 iterations with final norm 0.00017640556203934608

===== Begin Newton CG convergence information =====

=====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

```

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

||g||L2: l2 norm of gradient

alpha: step size

tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.5e+08	2.0e+07	----	5.0
e-01													
1	2	1	1	2	T	F	2.132606e+07	2.0e+07	1.5e+06	-3.7e+07	5.2e+06	1.0e+00	5.0
e-01													
2	2	1	1	2	T	F	3.338753e+06	2.8e+06	4.9e+05	-4.3e+06	1.5e+06	1.0e+00	2.8
e-01													
3	5	1	1	5	T	F	1.195733e+06	8.2e+05	3.7e+05	-1.0e+06	5.0e+05	1.0e+00	1.6
e-01													
4	7	1	1	7	T	F	6.670863e+05	3.7e+05	3.0e+05	-7.6e+04	8.5e+04	1.0e+00	6.6
e-02													
5	10	1	1	10	T	F	6.292693e+05	3.4e+05	2.9e+05	-4.5e+02	7.1e+03	1.0e+00	1.9
e-02													

```

 6 15 1 1 15 T F 6.290439e+05 3.4e+05 2.9e+05 -1.6e-01 1.3e+02 1.0e+00 2.6
e-03
 7 22 1 1 22 F F 6.290438e+05 3.4e+05 2.9e+05 -5.1e-05 1.2e+00 1.0e+00 2.5
e-04
 8 22 4 1 22 F F 6.290438e+05 3.4e+05 2.9e+05 -3.9e-05 1.0e+00 1.2e-01 2.3
e-04

```

```

converged : False
reason    : unknown reason
cumulative CG iterations : 86
cumulative cost evaluations : 12
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 86
===== End Newton CG convergence information =====
=====
```

```

===== Begin Newton CG convergence information =====
=====
```

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

||g||L2: l2 norm of gradient

alpha: step size

tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.5e+08	2.0e+07	----	5.0
1	2	1	1	2	T	F	2.132606e+07	2.0e+07	1.5e+06	-3.7e+07	5.2e+06	1.0e+00	5.0
2	2	1	1	2	T	F	3.338753e+06	2.8e+06	4.9e+05	-4.3e+06	1.5e+06	1.0e+00	2.8
3	5	1	1	5	T	F	1.195733e+06	8.2e+05	3.7e+05	-1.0e+06	5.0e+05	1.0e+00	1.6
4	7	1	1	7	T	F	6.670863e+05	3.7e+05	3.0e+05	-7.6e+04	8.5e+04	1.0e+00	6.6
5	10	1	1	10	T	F	6.292693e+05	3.4e+05	2.9e+05	-4.5e+02	7.1e+03	1.0e+00	1.9

```

 6 15 1 1 15 T F 6.290439e+05 3.4e+05 2.9e+05 -1.6e-01 1.3e+02 1.0e+00 2.6
e-03
 7 22 1 1 22 F F 6.290438e+05 3.4e+05 2.9e+05 -5.1e-05 1.2e+00 1.0e+00 2.5
e-04
 8 22 4 1 22 F F 6.290438e+05 3.4e+05 2.9e+05 -3.9e-05 1.0e+00 1.2e-01 2.3
e-04
 9 0 10 1 0 T F 6.290438e+05 3.4e+05 2.9e+05 ----- 1.0e+00 9.8e-04
----
```

```

converged : False
reason    : Maximum number of backtracking reached
cumulative CG iterations : 86
cumulative cost evaluations : 22
cumulative gradient evaluations : 10
cumulative Hessian vector products (excluding preconditioner builds) : 86
===== End Newton CG convergence information =====
=====
```

```
gamma= 5642.493956627751 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
827.8100435380474
```

```

----- gamma= 6118.097561746672 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format                = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 986413
sparsity constant = 20
size of H-matrix   = 4.06 MB
|A|_F                 = 986413
|S-A|_2 = 4.10017e-09
matrix has dimension 1416— H-matrix inverse ( rtol = 1e-06 , atol = 1e-14 ,
overwrite= False )
x 1416
no of non-zeroes      = 9746
matrix is            real valued
format                = symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 5.18561e+06
sparsity constant = 20
size of H-matrix   = 2.29 MB
|A|_F                 = 5.18561e+06
|S-A|_2 = 4.66653e-09
done in 14.772184371948242
size of inverse = 3361874 bytes
```

— H-matrix multiplication C=A*B

```
done in 24.07s
size of C = 3.87 MB
```

— H-matrix multiplication C=A*B

```
done in 32.68s
size of C = 8.54 MB
```

— LU factorisation (rtol = 1e-07)

```
done in 30.89s
size of LU factor = 18.69 MB
```

using Gauss-Newton Hessian

```
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
```

```

ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 6.138e+01
(M r0, r0) = 2.260e+04
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 27552264456638.92
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5249025.095828646

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 5.104e+06

```

```

||M r0|| = 5.441e+01
(M r0, r0) = 7.913e+03
Iteration : 0 (r, r) = 26054966978283.848
Iteration : 1 (r, r) = 6473242690748.515
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 2544256.805188603

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 2
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.058e+06
||M r0|| = 4.495e+01
(M r0, r0) = 5.731e+03

```

```

Iteration : 0 (r, r) = 9349885539661.662
Iteration : 1 (r, r) = 3913714972610.4346
Iteration : 2 (r, r) = 380462945413.51056
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 616816.7843156593

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 4
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 4
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.263e+05
||M r0|| = 2.282e+01

```

```

(M r0, r0) = 1.935e+03
Iteration : 0 (r, r) = 527449808817.5456
Iteration : 1 (r, r) = 160687550941.08383
Iteration : 2 (r, r) = 33790535932.055767
Iteration : 3 (r, r) = 18894419458.856133
Relative/Absolute residual less than tol
Converged in 3 iterations with final norm 137456.9731183403

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1  1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07  ---- 5.0
e-01
 1  1  1  1  1  T  F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01
 2  2  1  1  2  T  F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01
 3  3  1  1  3  T  F 1.727353e+06 1.2e+06 4.8e+05 -1.8e+06 7.3e+05 1.0e+00 1.9
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 7
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 7
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True

```

```

Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.390e+05
||M r0|| = 4.718e+00
(M r0, r0) = 4.313e+02
Iteration : 0 (r, r) = 19334059092.520103
Iteration : 1 (r, r) = 7247875720.422212
Iteration : 2 (r, r) = 4475951574.946624
Iteration : 3 (r, r) = 3452168534.8928995
Iteration : 4 (r, r) = 922591237.8618386
Iteration : 5 (r, r) = 215262624.86511877
Iteration : 6 (r, r) = 86070662.6460247
Relative/Absolute residual less than tol
Converged in 6 iterations with final norm 9277.427587754308

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1  1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07  ---- 5.0
e-01
 1  1  1  1  1  T  F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01
 2  2  1  1  2  T  F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01
 3  3  1  1  3  T  F 1.727353e+06 1.2e+06 4.8e+05 -1.8e+06 7.3e+05 1.0e+00 1.9
e-01
 4  6  1  1  6  T  F 7.894271e+05 4.4e+05 3.5e+05 -2.1e+05 1.4e+05 1.0e+00 8.4
e-02

converged : False
reason    : unknown reason
cumulative CG iterations : 13

```

```

cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 13
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.731e+04
||M r0|| = 4.250e+00
(M r0, r0) = 3.134e+02
Iteration : 0 (r, r) = 2238172375.0731626
Iteration : 1 (r, r) = 618700128.6462255
Iteration : 2 (r, r) = 87700054.27766621
Iteration : 3 (r, r) = 33430072.922515627
Iteration : 4 (r, r) = 19504510.562068425
Iteration : 5 (r, r) = 14327744.380251084
Iteration : 6 (r, r) = 7508999.932956887
Iteration : 7 (r, r) = 2390789.343373282
Relative/Absolute residual less than tol
Converged in 7 iterations with final norm 1546.2177541902959

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01

```

```

stokes_dev1

 2 2 1 1 2 T F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01
 3 3 1 1 3 T F 1.727353e+06 1.2e+06 4.8e+05 -1.8e+06 7.3e+05 1.0e+00 1.9
e-01
 4 6 1 1 6 T F 7.894271e+05 4.4e+05 3.5e+05 -2.1e+05 1.4e+05 1.0e+00 8.4
e-02
 5 7 1 1 7 T F 6.975534e+05 3.7e+05 3.3e+05 -3.8e+04 4.7e+04 1.0e+00 4.9
e-02

converged : False
reason    : unknown reason
cumulative CG iterations : 20
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 20
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.452e+03
||M r0|| = 3.327e-01
(M r0, r0) = 2.394e+01
Iteration : 0 (r, r) = 11917813.161069373
Iteration : 1 (r, r) = 4606513.007703627
Iteration : 2 (r, r) = 1420040.9352825829
Iteration : 3 (r, r) = 291530.37694936426
Iteration : 4 (r, r) = 146419.0735492895
Iteration : 5 (r, r) = 97358.04409870662
Iteration : 6 (r, r) = 48598.18128093131
Iteration : 7 (r, r) = 26698.38451811125
Iteration : 8 (r, r) = 6044.951726411782
Iteration : 9 (r, r) = 2235.356100178333
Iteration : 10 (r, r) = 1092.0475576731521
Relative/Absolute residual less than tol
Converged in 10 iterations with final norm 33.04614285621171

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

```

```

cost:    cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01
  1   1   1   1   1   T   F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01
  3   3   1   1   3   T   F 1.727353e+06 1.2e+06 4.8e+05 -1.8e+06 7.3e+05 1.0e+00 1.9
e-01
  4   6   1   1   6   T   F 7.894271e+05 4.4e+05 3.5e+05 -2.1e+05 1.4e+05 1.0e+00 8.4
e-02
  5   7   1   1   7   T   F 6.975534e+05 3.7e+05 3.3e+05 -3.8e+04 4.7e+04 1.0e+00 4.9
e-02
  6  10   1   1  10   T   F 6.786696e+05 3.5e+05 3.3e+05 -3.0e+02 3.5e+03 1.0e+00 1.3
e-02

converged : False
reason     : unknown reason
cumulative CG iterations : 30
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 30
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 6.775e+01
||M r0|| = 9.697e-03
(M r0, r0) = 6.108e-01
Iteration : 0 (r, r) = 4590.15483639251
Iteration : 1 (r, r) = 1664.681519007714
Iteration : 2 (r, r) = 443.28130339939287
Iteration : 3 (r, r) = 229.65525611157375
Iteration : 4 (r, r) = 90.54086617396655
Iteration : 5 (r, r) = 52.5888956772388
Iteration : 6 (r, r) = 19.89682549041862
Iteration : 7 (r, r) = 10.082156636912782
Iteration : 8 (r, r) = 4.29242688564112
Iteration : 9 (r, r) = 1.6595071118831572
Iteration : 10 (r, r) = 0.9519281920629445
Iteration : 11 (r, r) = 0.3710331930139767
Iteration : 12 (r, r) = 0.17407273758540226
Iteration : 13 (r, r) = 0.07348842838545781
Iteration : 14 (r, r) = 0.03208038019063672
Iteration : 15 (r, r) = 0.011814915820827995
Relative/Absolute residual less than tol
Converged in 15 iterations with final norm 0.10869643885991848

```

```
=====
Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01
  1   1   1   1   1   T   F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01
  3   3   1   1   3   T   F 1.727353e+06 1.2e+06 4.8e+05 -1.8e+06 7.3e+05 1.0e+00 1.9
e-01
  4   6   1   1   6   T   F 7.894271e+05 4.4e+05 3.5e+05 -2.1e+05 1.4e+05 1.0e+00 8.4
e-02
  5   7   1   1   7   T   F 6.975534e+05 3.7e+05 3.3e+05 -3.8e+04 4.7e+04 1.0e+00 4.9
e-02
  6   10  1   1   10  T   F 6.786696e+05 3.5e+05 3.3e+05 -3.0e+02 3.5e+03 1.0e+00 1.3
e-02
  7   15  1   1   15  F   F 6.785206e+05 3.5e+05 3.3e+05 -1.9e-01 6.8e+01 1.0e+00 1.9
e-03

converged : False
reason     : unknown reason
cumulative CG iterations : 45
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 45
=====
End Newton CG convergence information =====
=====

using Hessian
it= 8 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
```

```

lse
=====
===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 2.035642e+07 1.9e+07 1.5e+06 -2.6e+07 5.1e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 8.191928e+06 7.5e+06 7.3e+05 -1.3e+07 3.1e+06 1.0e+00 4.0
e-01
  3   3   1   1   3   T   F 1.727353e+06 1.2e+06 4.8e+05 -1.8e+06 7.3e+05 1.0e+00 1.9
e-01
  4   6   1   1   6   T   F 7.894271e+05 4.4e+05 3.5e+05 -2.1e+05 1.4e+05 1.0e+00 8.4
e-02
  5   7   1   1   7   T   F 6.975534e+05 3.7e+05 3.3e+05 -3.8e+04 4.7e+04 1.0e+00 4.9
e-02
  6   10  1   1   10  T   F 6.786696e+05 3.5e+05 3.3e+05 -3.0e+02 3.5e+03 1.0e+00 1.3
e-02
  7   15  1   1   15  F   F 6.785206e+05 3.5e+05 3.3e+05 -1.9e-01 6.8e+01 1.0e+00 1.9
e-03
  8   0   1   1   0   F   F 6.785205e+05 3.5e+05 3.3e+05      ---- 1.3e-01 1.0e+00
  ----

converged : True
reason    : Norm of the gradient less than tolerance
cumulative CG iterations : 45
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 45
===== End Newton CG convergence information =====
=====
```

```

gamma= 6118.097561746672 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
839.9312773370468

----- gamma= 7138.784456052863 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes = 9746
matrix is real valued
format = non symmetric
size of sparse matrix = 163.52 kB
|S|_F = 1.15098e+06
sparsity constant = 20
size of H-matrix = 4.06 MB
|A|_F = 1.15098e+06
|S-A|_2 = 6.6538e-09
matrix has dimension — H-matrix inverse ( rtol = 1e-06 , atol = 1e-14 , ove
rwrite= False )
done in 1416 x 1416
no of non-zeroes = 9746
matrix is real valued
format = symmetric
size of sparse matrix = 163.52 kB
|S|_F = 5.18561e+06
sparsity constant = 20
size of H-matrix = 2.29 MB
|A|_F = 5.18561e+06
|S-A|_2 = 4.66653e-09
14.250972747802734
size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
done in 23.86s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
done in 32.78s (5.27 GB)
size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
[=====using Gauss-Newton Hessian ] 7% E
TA 27 s (5.28 GB) ] 22% ETA 23 s (5.28 GB)
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
done in 30.47s
size of LU factor = 18.68 MB
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 5.778e+01
(M r0, r0) = 2.256e+04
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 26896234250609.027
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5186157.94694001

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p

```

```

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01

converged : False
reason      : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.915e+06
||M r0|| = 4.050e+01
(M r0, r0) = 7.108e+03
Iteration : 0 (r, r) = 24160913725032.47
Iteration : 1 (r, r) = 4453064542156.394
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 2110228.5521138213

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g

```

```

it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise
e.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 2
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.509e+06
||M r0|| = 3.539e+01
(M r0, r0) = 4.837e+03
Iteration : 0 (r, r) = 6296777088407.215
Iteration : 1 (r, r) = 2481593417039.141
Iteration : 2 (r, r) = 249372931029.08386
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 499372.5373196686

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g

```

```

...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.437744e+06 5.7e+06 7.8e+05 -9.9e+06 2.5e+06 1.0e+00 3.6
e-01

converged : False
reason      : unknown reason
cumulative CG iterations : 4
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 4
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M=approx= A^-1
||r0|| = 5.512e+05
||M r0|| = 1.881e+01
(M r0, r0) = 1.492e+03
  Iteration : 0 (r, r) = 303848102337.3596
  Iteration : 1 (r, r) = 81081111335.34378
  Iteration : 2 (r, r) = 15445353572.752089
  Iteration : 3 (r, r) = 18080127981.030334
  Iteration : 4 (r, r) = 8238387134.2983265
Relative/Absolute residual less than tol
Converged in 4 iterations with final norm 90765.56138920932

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.437744e+06 5.7e+06 7.8e+05 -9.9e+06 2.5e+06 1.0e+00 3.6
e-01
  3   4   1   1   4   T   F 1.512410e+06 9.8e+05 5.4e+05 -1.3e+06 5.5e+05 1.0e+00 1.7
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 8
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 8
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 9.127e+04
||M r0|| = 4.140e+00
(M r0, r0) = 3.319e+02
Iteration : 0 (r, r) = 8329932130.716492
Iteration : 1 (r, r) = 2496202487.435622
Iteration : 2 (r, r) = 532603679.47196174
Iteration : 3 (r, r) = 155225476.16986752
Iteration : 4 (r, r) = 92899021.41982985
Iteration : 5 (r, r) = 62839648.63868518
Iteration : 6 (r, r) = 36067250.26972498
Relative/Absolute residual less than tol
Converged in 6 iterations with final norm 6005.601574340823

```

```
=====
Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01
  1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.437744e+06 5.7e+06 7.8e+05 -9.9e+06 2.5e+06 1.0e+00 3.6
e-01
  3   4   1   1   4   T   F 1.512410e+06 9.8e+05 5.4e+05 -1.3e+06 5.5e+05 1.0e+00 1.7
e-01
  4   6   1   1   6   T   F 8.248326e+05 4.0e+05 4.2e+05 -6.5e+04 9.1e+04 1.0e+00 6.8
e-02

converged : False
reason     : unknown reason
cumulative CG iterations : 14
cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 14
=====
End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.603e+03
||M r0|| = 1.964e-01
(M r0, r0) = 2.055e+01
Iteration : 0 (r, r) = 57799774.423620336
```

```

Iteration : 1 (r, r) = 20459278.332493618
Iteration : 2 (r, r) = 5850681.833356286
Iteration : 3 (r, r) = 3141111.5110887783
Iteration : 4 (r, r) = 1860590.8383449935
Iteration : 5 (r, r) = 790604.9220607233
Iteration : 6 (r, r) = 226270.55212328077
Iteration : 7 (r, r) = 57373.93873669866
Iteration : 8 (r, r) = 13054.16419750683
Relative/Absolute residual less than tol
Converged in 8 iterations with final norm 114.25482133156058

```

```
===== Begin Newton CG convergence information =====
```

```
=====
Preconditioned inexact Newton-CG with line search
```

```
Hp=-g
```

```
u <- u + alpha * p
```

```
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

```
it:      Newton iteration number
```

```
nCG:     number of CG iterations in Newton iteration
```

```
nJ:      number of cost function evaluations in Newton iteration
```

```
nG:      number of gradient evaluations in Newton iteration
```

```
nHp:     number of Hessian-vector products in Newton iteration
```

```
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
```

```
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
```

```
cost:    cost, J = Jd + Jr
```

```
misfit:  misfit cost, Jd
```

```
reg:     regularization cost, Jr
```

```
(g,p):   inner product between gradient, g, and Newton search direction, p
```

```
||g||L2: l2 norm of gradient
```

```
alpha:   step size
```

```
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)
```

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0
e-01													
1	1	1	1	1	T	F	1.889649e+07	1.7e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0
e-01													
2	2	1	1	2	T	F	6.437744e+06	5.7e+06	7.8e+05	-9.9e+06	2.5e+06	1.0e+00	3.6
e-01													
3	4	1	1	4	T	F	1.512410e+06	9.8e+05	5.4e+05	-1.3e+06	5.5e+05	1.0e+00	1.7
e-01													
4	6	1	1	6	T	F	8.248326e+05	4.0e+05	4.2e+05	-6.5e+04	9.1e+04	1.0e+00	6.8
e-02													
5	8	1	1	8	T	F	7.925317e+05	3.8e+05	4.1e+05	-3.6e+02	7.6e+03	1.0e+00	2.0
e-02													

```
converged : False
```

```
reason    : unknown reason
```

```
cumulative CG iterations : 22
```

```
cumulative cost evaluations : 6
```

```
cumulative gradient evaluations : 6
```

```

cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.401e+02
||M r0|| = 8.849e-03
(M r0, r0) = 5.157e-01
Iteration : 0 (r, r) = 19638.404328884742
Iteration : 1 (r, r) = 8285.306905766436
Iteration : 2 (r, r) = 1890.3477045625775
Iteration : 3 (r, r) = 966.9571175218595
Iteration : 4 (r, r) = 401.7769648490362
Iteration : 5 (r, r) = 150.4335585114488
Iteration : 6 (r, r) = 83.94608334727351
Iteration : 7 (r, r) = 38.066598193353585
Iteration : 8 (r, r) = 13.989544974915455
Iteration : 9 (r, r) = 2.9651534590621953
Iteration : 10 (r, r) = 0.9213248930296841
Iteration : 11 (r, r) = 0.4743181490770158
Iteration : 12 (r, r) = 0.2159035316330078
Iteration : 13 (r, r) = 0.07901412561683538
Relative/Absolute residual less than tol
Converged in 13 iterations with final norm 0.28109451367259974

```

```

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
```

```

stokes_dev1
    0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07      ---- 5.0
e-01
    1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
    2   2   1   1   2   T   F 6.437744e+06 5.7e+06 7.8e+05 -9.9e+06 2.5e+06 1.0e+00 3.6
e-01
    3   4   1   1   4   T   F 1.512410e+06 9.8e+05 5.4e+05 -1.3e+06 5.5e+05 1.0e+00 1.7
e-01
    4   6   1   1   6   T   F 8.248326e+05 4.0e+05 4.2e+05 -6.5e+04 9.1e+04 1.0e+00 6.8
e-02
    5   8   1   1   8   T   F 7.925317e+05 3.8e+05 4.1e+05 -3.6e+02 7.6e+03 1.0e+00 2.0
e-02
    6  13   1   1  13   T   F 7.923532e+05 3.8e+05 4.2e+05 -1.9e-01 1.4e+02 1.0e+00 2.7
e-03

converged : False
reason     : unknown reason
cumulative CG iterations : 35
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 35
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.265e+00
||M r0|| = 1.335e-04
(M r0, r0) = 7.977e-03
Iteration : 0 (r, r) = 1.6008585910543833
Iteration : 1 (r, r) = 0.6159330822101328
Iteration : 2 (r, r) = 0.21619970253233936
Iteration : 3 (r, r) = 0.09148577901167894
Iteration : 4 (r, r) = 0.03977000800932326
Iteration : 5 (r, r) = 0.017194087226798146
Iteration : 6 (r, r) = 0.006674669191000224
Iteration : 7 (r, r) = 0.0021554660938514787
Iteration : 8 (r, r) = 0.0006252031774234368
Iteration : 9 (r, r) = 0.00022292934827211635
Iteration : 10 (r, r) = 9.29293999476769e-05
Iteration : 11 (r, r) = 3.998573440339819e-05
Iteration : 12 (r, r) = 1.718379362944741e-05
Iteration : 13 (r, r) = 7.231998152333773e-06
Iteration : 14 (r, r) = 2.429794614487861e-06
Iteration : 15 (r, r) = 8.638154780095389e-07
Iteration : 16 (r, r) = 4.104322110962867e-07
Iteration : 17 (r, r) = 1.7858321238271204e-07
Iteration : 18 (r, r) = 6.387157959502488e-08
Relative/Absolute residual less than tol
Converged in 18 iterations with final norm 0.00025272827225109756

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.437744e+06 5.7e+06 7.8e+05 -9.9e+06 2.5e+06 1.0e+00 3.6
e-01
  3   4   1   1   4   T   F 1.512410e+06 9.8e+05 5.4e+05 -1.3e+06 5.5e+05 1.0e+00 1.7
e-01
  4   6   1   1   6   T   F 8.248326e+05 4.0e+05 4.2e+05 -6.5e+04 9.1e+04 1.0e+00 6.8
e-02
  5   8   1   1   8   T   F 7.925317e+05 3.8e+05 4.1e+05 -3.6e+02 7.6e+03 1.0e+00 2.0
e-02
  6  13   1   1  13   T   F 7.923532e+05 3.8e+05 4.2e+05 -1.9e-01 1.4e+02 1.0e+00 2.7
e-03
  7  18   1   1  18   F   F 7.923531e+05 3.8e+05 4.2e+05 -4.3e-05 1.3e+00 1.0e+00 2.5
e-04

converged : False
reason    : unknown reason
cumulative CG iterations : 53
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====

using Hessian
it= 8 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct

```

ion

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:  l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01
  1   1   1   1   1   T   F 1.889649e+07 1.7e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.437744e+06 5.7e+06 7.8e+05 -9.9e+06 2.5e+06 1.0e+00 3.6
e-01
  3   4   1   1   4   T   F 1.512410e+06 9.8e+05 5.4e+05 -1.3e+06 5.5e+05 1.0e+00 1.7
e-01
  4   6   1   1   6   T   F 8.248326e+05 4.0e+05 4.2e+05 -6.5e+04 9.1e+04 1.0e+00 6.8
e-02
  5   8   1   1   8   T   F 7.925317e+05 3.8e+05 4.1e+05 -3.6e+02 7.6e+03 1.0e+00 2.0
e-02
  6  13   1   1  13   T   F 7.923532e+05 3.8e+05 4.2e+05 -1.9e-01 1.4e+02 1.0e+00 2.7
e-03
  7  18   1   1  18   F   F 7.923531e+05 3.8e+05 4.2e+05 -4.3e-05 1.3e+00 1.0e+00 2.5
e-04
  8   0   1   1   0   F   F 7.923531e+05 3.8e+05 4.2e+05   ---- 3.8e-02 1.0e+00
  ----

converged : True
reason     : Norm of the gradient less than tolerance
cumulative CG iterations : 53
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====
```

gamma= 7138.784456052863 , noise_datanorm= 863.5355316705534 , misfit_datanorm= 868.260089302643

----- gamma= 6959.888561201022 -----
 Building Regularization H-Matrix
 matrix has dimension 1416 x 1416
 no of non-zeroes = 9746

```

matrix is           real valued
format            = non symmetric
size of sparse matrix = 163.52 kB
|S|_F             = 1.12213e+06
sparsity constant = 20
size of H-matrix  = 4.06 MB
|A|_F             = 1.12213e+06
|S-A|_2 = 1.01053e-09
matrix has dimension — H-matrix inverse ( rtol = 1e-06 , atol = 1e-14 , ove
rwrite= False )
1416 x 1416
no of non-zeroes   = 9746
matrix is           real valued
format            = symmetric
size of sparse matrix = 163.52 kB
|S|_F             = 5.18561e+06
sparsity constant = 20
size of H-matrix  = 2.29 MB
|A|_F             = 5.18561e+06
|S-A|_2 = 4.66653e-09
14.947875022888184 ]    7% ETA 14 s (5.28 GB)
size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
using Gauss-Newton Hessian s (5.28 GB)
[===== () , num_gn_iter= 7 , using_gauss_newton=
True
done in 24.44s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
[===== Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 5.832e+01
(M r0, r0) = 2.257e+04
Iteration : 0 (r, r) = 380758959655851.75
1 (r, r) = 26994998919279.85B)4 s (5.28 GB)
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5195671.171203952

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nH:      number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis

```

```

e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

done in 33.03s                      100% ETA 0 s
(5.28 GB)
size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
                                         using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
done in 30.79s
size of LU factor = 18.69 MB
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.943e+06
||M r0|| = 4.248e+01
(M r0, r0) = 7.221e+03
Iteration : 0 (r, r) = 24437921378935.86
Iteration : 1 (r, r) = 4751109067522.971
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 2179703.894459743

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration

```

nHp: number of Hessian-vector products in Newton iteration
 GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
 BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
 cost: cost, $J = Jd + Jr$
 misfit: misfit cost, Jd
 reg: regularization cost, Jr
 (g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|L2$: l2 norm of gradient
 alpha: step size
 tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0e-01
1	1	1	1	1	T	F	1.910665e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0e-01

converged : False
 reason : unknown reason
 cumulative CG iterations : 2
 cumulative cost evaluations : 2
 cumulative gradient evaluations : 2
 cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information ======
=====

using Gauss-Newton Hessian
 it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
 Residual: r=b-Ax
 Preconditioner: M =approx= A^-1
 $\|r0\| = 2.598e+06$
 $\|M r0\| = 3.683e+01$
 $(M r0, r0) = 4.981e+03$
 Iteration : 0 (r, r) = 6751410678051.522
 Iteration : 1 (r, r) = 2681652363909.4595
 Iteration : 2 (r, r) = 263710772450.128
 Relative/Absolute residual less than tol
 Converged in 2 iterations with final norm 513527.7718391947

===== Begin Newton CG convergence information ======

=====
 Preconditioned inexact Newton-CG with line search
 Hp=-g
 $u \leftarrow u + \alpha * p$
 u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g
 it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
 ...
 it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number
 nCG: number of CG iterations in Newton iteration
 nJ: number of cost function evaluations in Newton iteration
 nG: number of gradient evaluations in Newton iteration
 nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
 BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
 cost: cost, J = Jd + Jr
 misfit: misfit cost, Jd
 reg: regularization cost, Jr
 (g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|_{L2}$: l2 norm of gradient
 alpha: step size
 tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01	1	1	1	1	T	F	1.910665e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.700183e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01														

converged : False
 reason : unknown reason
 cumulative CG iterations : 4
 cumulative cost evaluations : 3
 cumulative gradient evaluations : 3
 cumulative Hessian vector products (excluding preconditioner builds) : 4
 ===== End Newton CG convergence information ======
 =====

using Gauss-Newton Hessian
 it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
 Residual: r=b-Ax
 Preconditioner: M =approx= A^-1
 $\|r_0\| = 5.728e+05$
 $\|M r_0\| = 1.955e+01$
 $(M r_0, r_0) = 1.548e+03$
 Iteration : 0 (r, r) = 328139769359.62915
 Iteration : 1 (r, r) = 89739741084.64735
 Iteration : 2 (r, r) = 16224672247.429895
 Iteration : 3 (r, r) = 18226735419.339443
 Iteration : 4 (r, r) = 9990812893.898542
 Iteration : 5 (r, r) = 2721734252.176254
 Relative/Absolute residual less than tol
 Converged in 5 iterations with final norm 52170.24297601319

===== Begin Newton CG convergence information ======
 =====
 Preconditioned inexact Newton-CG with line search
 $Hp=-g$
 $u \leftarrow u + \alpha * p$
 u : parameter, J : cost, g : gradient, H : Hessian, α =step size, p =search direction
 it=0 : $u=u_0 \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $Hp=-g$
 it=1 : linesearch $u \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $Hp=-g$
 ...
 it=last : linesearch $u \rightarrow J \rightarrow g \rightarrow$ Done.
 it: Newton iteration number

nCG: number of CG iterations in Newton iteration
nJ: number of cost function evaluations in Newton iteration
nG: number of gradient evaluations in Newton iteration
nHp: number of Hessian-vector products in Newton iteration
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost: cost, $J = Jd + Jr$
misfit: misfit cost, Jd
reg: regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01	1	1	1	1	T	F	1.910665e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.700183e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.534167e+06	1.0e+06	5.2e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-01														

converged : False
reason : unknown reason
cumulative CG iterations : 9
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 9
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.209e+05
||M r0|| = 3.436e+00
(M r0, r0) = 4.296e+02
Iteration : 0 (r, r) = 14618524977.755209
Iteration : 1 (r, r) = 2824849143.357995
Iteration : 2 (r, r) = 596817292.8396115
Iteration : 3 (r, r) = 150264785.5796662
Iteration : 4 (r, r) = 47829032.384854436
Relative/Absolute residual less than tol
Converged in 4 iterations with final norm 6915.853698919204

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g

```

it=1      : linesearch u -> J -> g -> build precondition (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise
e.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

```

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	-----	5.0
e-01													
1	1	1	1	1	T	F	1.910665e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0
e-01													
2	2	1	1	2	T	F	6.700183e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01													
3	5	1	1	5	T	F	1.534167e+06	1.0e+06	5.2e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-01													
4	4	1	1	4	T	F	8.230069e+05	4.2e+05	4.0e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02													

```

converged : False
reason      : unknown reason
cumulative CG iterations : 13
cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 13
===== End Newton CG convergence information =====
=====
```

```

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.751e+03
||M r0|| = 2.715e-01
(M r0, r0) = 2.877e+01
Iteration : 0 (r, r) = 60084635.63848385
Iteration : 1 (r, r) = 29032064.085037418
Iteration : 2 (r, r) = 16385361.19183522
Iteration : 3 (r, r) = 9546030.446397679
Iteration : 4 (r, r) = 5178463.116675905
Iteration : 5 (r, r) = 1921669.6559056407
Iteration : 6 (r, r) = 700002.845715906
Iteration : 7 (r, r) = 224592.21862906194
Iteration : 8 (r, r) = 49460.683178396255
Iteration : 9 (r, r) = 15957.502524407959

```

```

Relative/Absolute residual less than tol
Converged in 9 iterations with final norm 126.32300868966017

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.910665e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.700183e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.534167e+06 1.0e+06 5.2e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
  4   4   1   1   4   T   F 8.230069e+05 4.2e+05 4.0e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
  5   9   1   1   9   T   F 7.721033e+05 3.7e+05 4.0e+05 -8.4e+02 7.8e+03 1.0e+00 2.0
e-02

converged : False
reason    : unknown reason
cumulative CG iterations : 22
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax

```

```

Preconditioner: M =approx= A^-1
||r0|| = 2.523e+02
||M r0|| = 8.286e-03
(M r0, r0) = 6.153e-01
Iteration : 0 (r, r) = 63644.77410180435
Iteration : 1 (r, r) = 18220.1013835282
Iteration : 2 (r, r) = 3418.748949646263
Iteration : 3 (r, r) = 2461.226163052246
Iteration : 4 (r, r) = 1117.9012560910874
Iteration : 5 (r, r) = 263.68204340349564
Iteration : 6 (r, r) = 111.33970591014665
Iteration : 7 (r, r) = 52.9728069177894
Iteration : 8 (r, r) = 15.931270925174996
Iteration : 9 (r, r) = 7.743578792046668
Iteration : 10 (r, r) = 2.612351129862421
Iteration : 11 (r, r) = 0.9782850927934768
Iteration : 12 (r, r) = 0.44378291462218056
Relative/Absolute residual less than tol
Converged in 12 iterations with final norm 0.66617033345407843

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07  ---- 5.0
e-01
 1  1  1  1   1  T  F 1.910665e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2  2  1  1   2  T  F 6.700183e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
 3  5  1  1   5  T  F 1.534167e+06 1.0e+06 5.2e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
 4  4  1  1   4  T  F 8.230069e+05 4.2e+05 4.0e+05 -1.0e+05 1.2e+05 1.0e+00 7.9


```

```

e-02
 5 9 1 1 9 T F 7.721033e+05 3.7e+05 4.0e+05 -8.4e+02 7.8e+03 1.0e+00 2.0
e-02
 6 12 1 1 12 T F 7.716808e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03

converged : False
reason    : unknown reason
cumulative CG iterations : 34
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 34
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.660e+00
||M r0|| = 1.342e-04
(M r0, r0) = 8.641e-03
  Iteration : 0 (r, r) = 2.7544941060569483
  Iteration : 1 (r, r) = 1.176639982063275
  Iteration : 2 (r, r) = 0.4726279462324404
  Iteration : 3 (r, r) = 0.2303162865384366
  Iteration : 4 (r, r) = 0.13953013709331114
  Iteration : 5 (r, r) = 0.05438198365811223
  Iteration : 6 (r, r) = 0.019923023823623057
  Iteration : 7 (r, r) = 0.00717110597089139
  Iteration : 8 (r, r) = 0.0028992477866755053
  Iteration : 9 (r, r) = 0.001446070443569261
  Iteration : 10 (r, r) = 0.0007244671087460257
  Iteration : 11 (r, r) = 0.0002932677316112949
  Iteration : 12 (r, r) = 0.00010507512280767351
  Iteration : 13 (r, r) = 3.72073125707393e-05
  Iteration : 14 (r, r) = 1.1765484101651073e-05
  Iteration : 15 (r, r) = 4.322235664123169e-06
  Iteration : 16 (r, r) = 1.4039536226039814e-06
  Iteration : 17 (r, r) = 5.527179294902568e-07
  Iteration : 18 (r, r) = 2.5110672677278734e-07
  Iteration : 19 (r, r) = 7.873527934973784e-08
Relative/Absolute residual less than tol
Converged in 19 iterations with final norm 0.000280598074387081

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration

```

nJ: number of cost function evaluations in Newton iteration
 nG: number of gradient evaluations in Newton iteration
 nHp: number of Hessian-vector products in Newton iteration
 GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
 BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
 cost: cost, $J = J_d + J_r$
 misfit: misfit cost, J_d
 reg: regularization cost, J_r
 (g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|_{L2}$: l2 norm of gradient
 alpha: step size
 tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01	1	1	1	1	T	F	1.910665e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.700183e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.534167e+06	1.0e+06	5.2e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-01	4	4	1	1	4	T	F	8.230069e+05	4.2e+05	4.0e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02	5	9	1	1	9	T	F	7.721033e+05	3.7e+05	4.0e+05	-8.4e+02	7.8e+03	1.0e+00	2.0
e-02	6	12	1	1	12	T	F	7.716808e+05	3.7e+05	4.0e+05	-3.3e-01	2.5e+02	1.0e+00	3.6
e-03	7	19	1	1	19	F	F	7.716807e+05	3.7e+05	4.0e+05	-5.9e-05	1.7e+00	1.0e+00	2.9
e-04														

converged : False
 reason : unknown reason
 cumulative CG iterations : 53
 cumulative cost evaluations : 8
 cumulative gradient evaluations : 8
 cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information ======
=====

using Hessian
 it= 8 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= False

===== Begin Newton CG convergence information ======
=====

Preconditioned inexact Newton-CG with line search

$H_p = -g$

$u \leftarrow u + \alpha * p$

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve $H_p = -g$
 it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve $H_p = -g$
 ...
 it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration
nJ: number of cost function evaluations in Newton iteration
nG: number of gradient evaluations in Newton iteration
nHp: number of Hessian-vector products in Newton iteration
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost: cost, $J = J_d + J_r$
misfit: misfit cost, J_d
reg: regularization cost, J_r
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01	1	1	1	1	T	F	1.910665e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.700183e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.534167e+06	1.0e+06	5.2e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-02	4	4	1	1	4	T	F	8.230069e+05	4.2e+05	4.0e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02	5	9	1	1	9	T	F	7.721033e+05	3.7e+05	4.0e+05	-8.4e+02	7.8e+03	1.0e+00	2.0
e-03	6	12	1	1	12	T	F	7.716808e+05	3.7e+05	4.0e+05	-3.3e-01	2.5e+02	1.0e+00	3.6
e-04	7	19	1	1	19	F	F	7.716807e+05	3.7e+05	4.0e+05	-5.9e-05	1.7e+00	1.0e+00	2.9
	8	0	1	1	0	F	F	7.716807e+05	3.7e+05	4.0e+05	-----	3.9e-02	1.0e+00	

converged : True
reason : Norm of the gradient less than tolerance
cumulative CG iterations : 53
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====

gamma= 6959.888561201022 , noise_datanorm= 863.5355316705534 , misfit_datanorm= 863.0787236500014

----- gamma= 6975.521475652478 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes = 9746
matrix is real valued
format = non symmetric
size of sparse matrix = 163.52 kB
|S|_F = 1.12465e+06
sparsity constant = 20
size of H-matrix = 4.06 MB
|A|_F = 1.12465e+06
|S-A|_2 = 3.50572e-09

```

matrix has dimension — H-matrix inverse ( rtol = 1e-06 , atol = 1e-14 , ove
rwrite= False )
1416 x 1416
    no of non-zeroes      = 9746
    matrix is            real valued
    format                = symmetric
    size of sparse matrix = 163.52 kB
    |S|_F                 = 5.18561e+06
    sparsity constant = 20
    size of H-matrix   = 2.29 MB
    |A|_F                 = 5.18561e+06
|S-A|_2 = 4.66653e-09
done in 14.838630676269531
size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
done in 23.15s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
done in 32.63s
size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
[=====] 52using Gauss-Newton Hessian
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
[=====] 82% ETA 5 s Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 5.828e+01
(M r0, r0) = 2.257e+04
Iteration : 0 (r, r) = 380758959655851.75
    done in 30.54s
    size of LU factor = 18.68 MB
Iteration : 1 (r, r) = 26986137540882.38
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5194818.335695906

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr

```

```

misfit: misfit cost, Jd
reg: regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg   (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01

converged : False
reason      : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.941e+06
||M r0|| = 4.230e+01
(M r0, r0) = 7.211e+03
Iteration : 0 (r, r) = 24412943490066.926
Iteration : 1 (r, r) = 4724179467642.846
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 2173517.7633603197

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0     : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1     : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient

```

```

alpha: step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost misfit reg (g,p) ||g||L2 alpha t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07 ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908764e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 2
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.590e+06
||M r0|| = 3.670e+01
(M r0, r0) = 4.968e+03
Iteration : 0 (r, r) = 6710410662649.094
Iteration : 1 (r, r) = 2663424519425.3853
Iteration : 2 (r, r) = 262383989939.66464
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 512234.3115603099

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size

```

tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	-----	5.0	
e-01	1	1	1	1	T	F	1.908764e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.676546e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01														

converged : False

reason : unknown reason

cumulative CG iterations : 4

cumulative cost evaluations : 3

cumulative gradient evaluations : 3

cumulative Hessian vector products (excluding preconditioner builds) : 4

===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian

it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True

Residual: $r=b-Ax$

Preconditioner: M =approx= A^{-1}

$\|r_0\| = 5.708e+05$

$\|M r_0\| = 1.948e+01$

$(M r_0, r_0) = 1.543e+03$

Iteration : 0 $(r, r) = 325862045174.4467$

Iteration : 1 $(r, r) = 88927921774.37445$

Iteration : 2 $(r, r) = 16139933250.665852$

Iteration : 3 $(r, r) = 18219006083.39456$

Iteration : 4 $(r, r) = 9807126242.911156$

Iteration : 5 $(r, r) = 2640484583.114731$

Relative/Absolute residual less than tol

Converged in 5 iterations with final norm 51385.64569132834

===== Begin Newton CG convergence information =====
=====

Preconditioned inexact Newton-CG with line search

$H_p = -g$

$u \leftarrow u + \alpha * p$

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : $u=u_0 \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $H_p = -g$
it=1 : linesearch $u \rightarrow J \rightarrow g \rightarrow$ build precond (optional) \rightarrow cgssolve $H_p = -g$

...

it=last : linesearch $u \rightarrow J \rightarrow g \rightarrow$ Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, $J = J_d + J_r$

misfit: misfit cost, J_d

```

reg:      regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg   (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908764e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.676546e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.532065e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01

converged : False
reason   : unknown reason
cumulative CG iterations : 9
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 9
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.207e+05
||M r0|| = 3.452e+00
(M r0, r0) = 4.292e+02
Iteration : 0 (r, r) = 14560597816.619476
Iteration : 1 (r, r) = 2744621290.539221
Iteration : 2 (r, r) = 590918825.7678332
Iteration : 3 (r, r) = 146934718.8114643
Iteration : 4 (r, r) = 46961452.457745984
Relative/Absolute residual less than tol
Converged in 4 iterations with final norm 6852.842655259639

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0     : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1     : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration

```

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
 BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
 cost: cost, $J = Jd + Jr$
 misfit: misfit cost, Jd
 reg: regularization cost, Jr
 (g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|L2$: l2 norm of gradient
 alpha: step size
 tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01	1	1	1	1	T	F	1.908764e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.676546e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.532065e+06	1.0e+06	5.3e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-02	4	4	1	1	4	T	F	8.249513e+05	4.2e+05	4.1e+05	-1.0e+05	1.2e+05	1.0e+00	7.9

converged : False
 reason : unknown reason
 cumulative CG iterations : 13
 cumulative cost evaluations : 5
 cumulative gradient evaluations : 5
 cumulative Hessian vector products (excluding preconditioner builds) : 13
 ===== End Newton CG convergence information ======

using Gauss-Newton Hessian
 it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
 Residual: r=b-Ax
 Preconditioner: M =approx= A^-1
 $\|r0\| = 7.731e+03$
 $\|M r0\| = 2.759e-01$
 $(M r0, r0) = 2.884e+01$
 Iteration : 0 (r, r) = 59764721.66030115
 Iteration : 1 (r, r) = 28736506.922672182
 Iteration : 2 (r, r) = 16147166.295022048
 Iteration : 3 (r, r) = 9345878.847465008
 Iteration : 4 (r, r) = 5078625.805474104
 Iteration : 5 (r, r) = 1900877.6928434072
 Iteration : 6 (r, r) = 686175.4882842426
 Iteration : 7 (r, r) = 222592.27080003245
 Iteration : 8 (r, r) = 49276.8067354832
 Iteration : 9 (r, r) = 16046.743004559743
 Relative/Absolute residual less than tol
 Converged in 9 iterations with final norm 126.67573960533936

===== Begin Newton CG convergence information ======

=====

Preconditioned inexact Newton-CG with line search
 $Hp=-g$
 $u \leftarrow u + \alpha * p$
 u : parameter, J : cost, g : gradient, H : Hessian, α =step size, p =search direct

```

ion

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):     inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908764e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.676546e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.532065e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
  4   4   1   1   4   T   F 8.249513e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
  5   9   1   1   9   T   F 7.738958e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02

converged : False
reason      : unknown reason
cumulative CG iterations : 22
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.492e+02
||M r0|| = 8.120e-03
(M r0, r0) = 6.144e-01
Iteration : 0 (r, r) = 62079.25217709316
Iteration : 1 (r, r) = 17820.525276786488
Iteration : 2 (r, r) = 3329.24037129121
Iteration : 3 (r, r) = 2346.838581740703
Iteration : 4 (r, r) = 1071.5360812891247

```

```

Iteration : 5 (r, r) = 250.77167391711976
Iteration : 6 (r, r) = 107.58866544384756
Iteration : 7 (r, r) = 51.31551518144478
Iteration : 8 (r, r) = 15.597486739681184
Iteration : 9 (r, r) = 7.515118900809812
Iteration : 10 (r, r) = 2.498509297210107
Iteration : 11 (r, r) = 0.9220808618184567
Iteration : 12 (r, r) = 0.41492071630487126
Relative/Absolute residual less than tol
Converged in 12 iterations with final norm 0.6441433973152805

```

```
===== Begin Newton CG convergence information =====
```

```
=====
Preconditioned inexact Newton-CG with line search
```

```
Hp=-g
```

```
u <- u + alpha * p
```

```
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

```
it:      Newton iteration number
```

```
nCG:     number of CG iterations in Newton iteration
```

```
nJ:      number of cost function evaluations in Newton iteration
```

```
nG:      number of gradient evaluations in Newton iteration
```

```
nHp:     number of Hessian-vector products in Newton iteration
```

```
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
```

```
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
```

```
cost:    cost, J = Jd + Jr
```

```
misfit:  misfit cost, Jd
```

```
reg:     regularization cost, Jr
```

```
(g,p):   inner product between gradient, g, and Newton search direction, p
```

```
||g||L2: l2 norm of gradient
```

```
alpha:   step size
```

```
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)
```

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0
e-01	1	1	1	1	T	F	1.908764e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0
e-01	2	2	1	1	2	T	6.676546e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	1.532065e+06	1.0e+06	5.3e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-01	4	4	1	1	4	T	8.249513e+05	4.2e+05	4.1e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02	5	9	1	1	9	T	7.738958e+05	3.7e+05	4.0e+05	-8.4e+02	7.7e+03	1.0e+00	2.0
e-02	6	12	1	1	12	T	7.734754e+05	3.7e+05	4.0e+05	-3.3e-01	2.5e+02	1.0e+00	3.6
e-03													

```
converged : False
```

```
reason    : unknown reason
```

```
cumulative CG iterations : 34
```

```

cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 34
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.626e+00
||M r0|| = 1.314e-04
(M r0, r0) = 8.542e-03
Iteration : 0 (r, r) = 2.6447626949387555
Iteration : 1 (r, r) = 1.1047201600059233
Iteration : 2 (r, r) = 0.4386994619299106
Iteration : 3 (r, r) = 0.21167325627094197
Iteration : 4 (r, r) = 0.1293118812986188
Iteration : 5 (r, r) = 0.05214425783584359
Iteration : 6 (r, r) = 0.019147589746999066
Iteration : 7 (r, r) = 0.006863200589848971
Iteration : 8 (r, r) = 0.002727831677560976
Iteration : 9 (r, r) = 0.001331467839776729
Iteration : 10 (r, r) = 0.0006641766947584329
Iteration : 11 (r, r) = 0.00027318130329950975
Iteration : 12 (r, r) = 9.61178228052334e-05
Iteration : 13 (r, r) = 3.439319736998566e-05
Iteration : 14 (r, r) = 1.0598821216691e-05
Iteration : 15 (r, r) = 3.9056143787202344e-06
Iteration : 16 (r, r) = 1.2832964185919377e-06
Iteration : 17 (r, r) = 5.026065464888799e-07
Iteration : 18 (r, r) = 2.2586032906766434e-07
Iteration : 19 (r, r) = 7.20061079874386e-08
Relative/Absolute residual less than tol
Converged in 19 iterations with final norm 0.00026833953862120025

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr

```

(g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|_{L2}$: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	-----	5.0	
e-01	1	1	1	1	T	F	1.908764e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.676546e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.532065e+06	1.0e+06	5.3e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-02	4	4	1	1	4	T	F	8.249513e+05	4.2e+05	4.1e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02	5	9	1	1	9	T	F	7.738958e+05	3.7e+05	4.0e+05	-8.4e+02	7.7e+03	1.0e+00	2.0
e-03	6	12	1	1	12	T	F	7.734754e+05	3.7e+05	4.0e+05	-3.3e-01	2.5e+02	1.0e+00	3.6
e-04	7	19	1	1	19	F	F	7.734753e+05	3.7e+05	4.0e+05	-5.7e-05	1.6e+00	1.0e+00	2.9

converged : False
reason : unknown reason
cumulative CG iterations : 53
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information ======

===== Begin Newton CG convergence information ======

=====
Preconditioned inexact Newton-CG with line search
 $Hp=-g$
 $u \leftarrow u + \alpha * p$
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve $Hp=-g$
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve $Hp=-g$
...
it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number
nCG: number of CG iterations in Newton iteration
nJ: number of cost function evaluations in Newton iteration
nG: number of gradient evaluations in Newton iteration
nHp: number of Hessian-vector products in Newton iteration
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost: cost, $J = Jd + Jr$
misfit: misfit cost, Jd
reg: regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|_{L2}$: l2 norm of gradient

```

alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost misfit reg (g,p) ||g||L2 alpha t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07 ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908764e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.676546e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.532065e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
  4   4   1   1   4   T   F 8.249513e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
  5   9   1   1   9   T   F 7.738958e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02
  6  12   1   1  12   T   F 7.734754e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03
  7  19   1   1  19   F   F 7.734753e+05 3.7e+05 4.0e+05 -5.7e-05 1.6e+00 1.0e+00 2.9
e-04
  8   0  10   1   0   T   F 7.734753e+05 3.7e+05 4.0e+05 ---- 1.6e+00 9.8e-04
  ----

converged : False
reason   : Maximum number of backtracking reached
cumulative CG iterations : 53
cumulative cost evaluations : 18
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====

gamma= 6975.521475652478 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
863.5279664560159

----- gamma= 6975.784963434012 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format               = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                = 1.1247e+06
sparsity constant = 20
size of H-matrix   = 4.06 MB
|A|_F                = 1.1247e+06
|S-A|_2 = 6.53779e-09
matrix has dimension — H-matrix inverse ( rtol = 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format               = symmetric
size of sparse matrix = 163.52 kB
|S|_F                = 5.18561e+06
sparsity constant = 20
size of H-matrix   = 2.29 MB
|A|_F                = 5.18561e+06
|S-A|_2 = 4.66653e-09
1e-06 , atol = 1e-14 , overwrite= False )
done in 14.96683955192566

```

```

size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
done in 24.33s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
done in 33.59s
size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
[=====
] 37% ETA 19 s (5.28 GB)===== ] 52% ETA 14 s (5.
28 GB)
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
done in 30.85s
size of LU factor = 18.68 MB
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 5.827e+01
(M r0, r0) = 2.257e+04
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 26985988575434.09
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5194803.997787991

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number
nCG: number of CG iterations in Newton iteration
nJ: number of cost function evaluations in Newton iteration
nG: number of gradient evaluations in Newton iteration
nHp: number of Hessian-vector products in Newton iteration
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost: cost, J = Jd + Jr
misfit: misfit cost, Jd
reg: regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)

it nCG nJ nG nHp GN BP cost misfit reg (g,p) ||g||L2 alpha t
olcg
0 1 1 1 1 T F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07 ---- 5.0

```

e-01

```

converged : False
reason    : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.941e+06
||M r0|| = 4.230e+01
(M r0, r0) = 7.211e+03
Iteration : 0 (r, r) = 24412523831736.195
Iteration : 1 (r, r) = 4723727061156.914
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 2173413.6884534694

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F  2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F  1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01

```

```

converged : False
reason    : unknown reason
cumulative CG iterations : 2
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.590e+06
||M r0|| = 3.670e+01
(M r0, r0) = 4.968e+03
Iteration : 0 (r, r) = 6709721660493.758
Iteration : 1 (r, r) = 2663118516506.841
Iteration : 2 (r, r) = 262361761555.70374
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 512212.6136241705

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
 1   1   1   1   1   T   F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2   2   1   1   2   T   F 6.676149e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01

```

```

converged : False
reason    : unknown reason
cumulative CG iterations : 4
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 4
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 5.708e+05
||M r0|| = 1.948e+01
(M r0, r0) = 1.543e+03
Iteration : 0 (r, r) = 325823942410.9524
Iteration : 1 (r, r) = 88914339428.99207
Iteration : 2 (r, r) = 16138539026.315361
Iteration : 3 (r, r) = 18218870484.611763
Iteration : 4 (r, r) = 9804070321.366941
Iteration : 5 (r, r) = 2639133370.625786
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 51372.49624678351

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01

```

```

 1  1  1  1   1  T  F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2  2  1  1   2  T  F 6.676149e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
 3  5  1  1   5  T  F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01

```

```

converged : False
reason    : unknown reason
cumulative CG iterations : 9
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 9
===== End Newton CG convergence information =====
=====
```

```

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.207e+05
||M r0|| = 3.452e+00
(M r0, r0) = 4.292e+02
Iteration : 0 (r, r) = 14559585462.508945
Iteration : 1 (r, r) = 2743233587.129353
Iteration : 2 (r, r) = 590817306.181736
Iteration : 3 (r, r) = 146878827.28704348
Iteration : 4 (r, r) = 46947078.0784325
Relative/Absolute residual less than tol
Converged in 4 iterations with final norm 6851.793785457389
```

```

===== Begin Newton CG convergence information =====
=====
```

```

Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion
```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

```

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)
```

```

it nCG nJ nG nHp GN BP          cost  misfit      reg  (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.676149e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
  4   4   1   1   4   T   F 8.249840e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02

```

```

converged : False
reason    : unknown reason
cumulative CG iterations : 13
cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 13
===== End Newton CG convergence information =====
=====

```

```

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Tr
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.730e+03
||M r0|| = 2.760e-01
(M r0, r0) = 2.884e+01
  Iteration : 0 (r, r) = 59759317.696027935
  Iteration : 1 (r, r) = 28731720.293041974
  Iteration : 2 (r, r) = 16143262.141422078
  Iteration : 3 (r, r) = 9342633.197015144
  Iteration : 4 (r, r) = 5076984.054706373
  Iteration : 5 (r, r) = 1900521.2619439233
  Iteration : 6 (r, r) = 685941.8592473419
  Iteration : 7 (r, r) = 222556.75456658046
  Iteration : 8 (r, r) = 49272.61862664743
  Iteration : 9 (r, r) = 16047.785900776398
Relative/Absolute residual less than tol
Converged in 9 iterations with final norm 126.67985593919974

```

```

===== Begin Newton CG convergence information =====
=====

```

```

Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

```

```

it=0     : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1     : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

```

```

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration

```

nHp: number of Hessian-vector products in Newton iteration
 GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
 BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
 cost: cost, $J = Jd + Jr$
 misfit: misfit cost, Jd
 reg: regularization cost, Jr
 (g,p): inner product between gradient, g, and Newton search direction, p
 $\|g\|L2$: l2 norm of gradient
 alpha: step size
 tolcg: relative tolerance for $Hp=-g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	-----	5.0	
e-01	1	1	1	1	T	F	1.908732e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.676149e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.532030e+06	1.0e+06	5.3e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-02	4	4	1	1	4	T	F	8.249840e+05	4.2e+05	4.1e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02	5	9	1	1	9	T	F	7.739261e+05	3.7e+05	4.0e+05	-8.4e+02	7.7e+03	1.0e+00	2.0

converged : False
 reason : unknown reason
 cumulative CG iterations : 22
 cumulative cost evaluations : 6
 cumulative gradient evaluations : 6
 cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information ======
=====

using Gauss-Newton Hessian
 it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
 Residual: r=b-Ax
 Preconditioner: M =approx= A^-1
 $\|r0\| = 2.491e+02$
 $\|M r0\| = 8.117e-03$
 $(M r0, r0) = 6.144e-01$
 Iteration : 0 (r, r) = 62058.439250244424
 Iteration : 1 (r, r) = 17814.407700301672
 Iteration : 2 (r, r) = 3327.902969464403
 Iteration : 3 (r, r) = 2344.9836647559982
 Iteration : 4 (r, r) = 1070.8140537124389
 Iteration : 5 (r, r) = 250.56244329211685
 Iteration : 6 (r, r) = 107.52763895864862
 Iteration : 7 (r, r) = 51.28823507945161
 Iteration : 8 (r, r) = 15.591701323935272
 Iteration : 9 (r, r) = 7.511493688747701
 Iteration : 10 (r, r) = 2.496720934047522
 Iteration : 11 (r, r) = 0.921224661109163
 Iteration : 12 (r, r) = 0.4144705377550145
 Relative/Absolute residual less than tol
 Converged in 12 iterations with final norm 0.6437938627814143

```
=====
Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07   ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.676149e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
  4   4   1   1   4   T   F 8.249840e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
  5   9   1   1   9   T   F 7.739261e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02
  6  12   1   1  12   T   F 7.735057e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03

converged : False
reason    : unknown reason
cumulative CG iterations : 34
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 34
=====
End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= False
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
```

```

||r0|| = 1.623e+00
||M r0|| = 1.314e-04
(M r0, r0) = 8.544e-03
Iteration : 0 (r, r) = 2.633967188440769
Iteration : 1 (r, r) = 1.1024529077736012
Iteration : 2 (r, r) = 0.4386488845931381
Iteration : 3 (r, r) = 0.21238073485041603
Iteration : 4 (r, r) = 0.12948340181010617
Iteration : 5 (r, r) = 0.052230074546741716
Iteration : 6 (r, r) = 0.019163275521390938
Iteration : 7 (r, r) = 0.006856352729135722
Iteration : 8 (r, r) = 0.00272804091874754
Iteration : 9 (r, r) = 0.001331541590609222
Iteration : 10 (r, r) = 0.0006644383964490574
Iteration : 11 (r, r) = 0.00027234016006245855
Iteration : 12 (r, r) = 9.599314296330871e-05
Iteration : 13 (r, r) = 3.4389019181241365e-05
Iteration : 14 (r, r) = 1.0581667883611036e-05
Iteration : 15 (r, r) = 3.892591692362938e-06
Iteration : 16 (r, r) = 1.2833389844066836e-06
Iteration : 17 (r, r) = 5.028329845578269e-07
Iteration : 18 (r, r) = 2.2534349144204616e-07
Iteration : 19 (r, r) = 7.206965224998767e-08
Relative/Absolute residual less than tol
Converged in 19 iterations with final norm 0.0002684579152306515

```

```

=====
 Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
 1  1  1  1   1  T  F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0

```

```

e-01
 2 2 1 1 2 T F 6.676149e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
 3 5 1 1 5 T F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
 4 4 1 1 4 T F 8.249840e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
 5 9 1 1 9 T F 7.739261e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02
 6 12 1 1 12 T F 7.735057e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03
 7 19 1 1 19 F F 7.735055e+05 3.7e+05 4.0e+05 -5.7e-05 1.6e+00 1.0e+00 2.9
e-04

converged : False
reason    : unknown reason
cumulative CG iterations : 53
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====

using Hessian
it= 8 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0 1 1 1 1 T F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01

```

```

 1  1  1  1   1  T  F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2  2  1  1   2  T  F 6.676149e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
 3  5  1  1   5  T  F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
 4  4  1  1   4  T  F 8.249840e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
 5  9  1  1   9  T  F 7.739261e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02
 6  12 1  1  12  T  F 7.735057e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03
 7  19 1  1  19  F  F 7.735055e+05 3.7e+05 4.0e+05 -5.7e-05 1.6e+00 1.0e+00 2.9
e-04
 8  0  1  1   0  F  F 7.735055e+05 3.7e+05 4.0e+05      ---- 3.6e-02 1.0e+00
----
```

```

converged : True
reason    : Norm of the gradient less than tolerance
cumulative CG iterations : 53
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====
```

```
gamma= 6975.784963434012 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
863.5355515346085
```

```

----- gamma= 6975.784273393516 -----
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format                = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 1.1247e+06
sparsity constant = 20
size of H-matrix   = 4.06 MB
|A|_F                 = 1.1247e+06
|S-A|_2 = 7.02535e-10
matrix has dimension — H-matrix inverse ( rtol = 1416 x 1416
no of non-zeroes      = 9746
matrix is            real valued
format                = symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 5.18561e+06
sparsity constant = 20
size of H-matrix   = 2.29 MB
|A|_F                 = 5.18561e+06
|S-A|_2 = 4.66653e-09
1e-06 , atol = 1e-14 , overwrite= False )
done in 14.302937030792236
size of inverse = 3361874 bytes
```

— H-matrix multiplication C=A*B

done in 23.91s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
done in 32.89s

```

size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
done in 30.35s
size of LU factor = 18.69 MB
using Gauss-Newton Hessian
it= 0 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 5.827e+01
(M r0, r0) = 2.257e+04
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 26985988965503.54
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 5194804.035332183

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====
```

```

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.941e+06
||M r0|| = 4.230e+01
(M r0, r0) = 7.211e+03
Iteration : 0 (r, r) = 24412524966420.227
Iteration : 1 (r, r) = 4723728264318.135
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 2173413.965244112

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1  1  1  1   1  T  F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 2
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 2
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian

```

```

it= 2 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
ue
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.590e+06
||M r0|| = 3.670e+01
(M r0, r0) = 4.968e+03
Iteration : 0 (r, r) = 6709723502914.271
Iteration : 1 (r, r) = 2663119337060.9033
Iteration : 2 (r, r) = 262361819223.133
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 512212.6699166402

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha  t
olcg
 0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
 1   1   1   1   1   T   F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2   2   1   1   2   T   F 6.676150e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 4
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 4
===== End Newton CG convergence information =====
=====
```

```

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 5.708e+05
||M r0|| = 1.948e+01
(M r0, r0) = 1.543e+03
Iteration : 0 (r, r) = 325824031054.1389
Iteration : 1 (r, r) = 88914373127.82126
Iteration : 2 (r, r) = 16138542638.227646
Iteration : 3 (r, r) = 18218870739.925636
Iteration : 4 (r, r) = 9804078273.521988
Iteration : 5 (r, r) = 2639136942.956028
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 51372.5310156705

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F  2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
  1   1   1   1   1   T   F  1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F  6.676150e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F  1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 9

```

```

cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 9
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.207e+05
||M r0|| = 3.452e+00
(M r0, r0) = 4.292e+02
Iteration : 0 (r, r) = 14559587774.812725
Iteration : 1 (r, r) = 2743237233.9661226
Iteration : 2 (r, r) = 590817485.3079911
Iteration : 3 (r, r) = 146878956.0452265
Iteration : 4 (r, r) = 46947112.831418544
Relative/Absolute residual less than tol
Converged in 4 iterations with final norm 6851.796321507123

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha    t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07    ---- 5.0
e-01
 1  1  1  1   1  T  F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2  2  1  1   2  T  F 6.676150e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
 3  5  1  1   5  T  F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7


```

```

e-01
 4 4 1 1 4 T F 8.249839e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02

converged : False
reason    : unknown reason
cumulative CG iterations : 13
cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 13
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.730e+03
||M r0|| = 2.760e-01
(M r0, r0) = 2.884e+01
  Iteration : 0 (r, r) = 59759315.870615825
  Iteration : 1 (r, r) = 28731736.060458593
  Iteration : 2 (r, r) = 16143277.103502933
  Iteration : 3 (r, r) = 9342645.571804605
  Iteration : 4 (r, r) = 5076988.515400544
  Iteration : 5 (r, r) = 1900522.3241589246
  Iteration : 6 (r, r) = 685942.5012781622
  Iteration : 7 (r, r) = 222556.85340325884
  Iteration : 8 (r, r) = 49272.65866337853
  Iteration : 9 (r, r) = 16047.785113285427
Relative/Absolute residual less than tol
Converged in 9 iterations with final norm 126.67985283100634

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size

```

tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ L2$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0	
e-01	1	1	1	1	T	F	1.908732e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0	
e-01	2	2	1	1	2	T	F	6.676150e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01	3	5	1	1	5	T	F	1.532030e+06	1.0e+06	5.3e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-01	4	4	1	1	4	T	F	8.249839e+05	4.2e+05	4.1e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02	5	9	1	1	9	T	F	7.739260e+05	3.7e+05	4.0e+05	-8.4e+02	7.7e+03	1.0e+00	2.0
e-02														

converged : False

reason : unknown reason

cumulative CG iterations : 22

cumulative cost evaluations : 6

cumulative gradient evaluations : 6

cumulative Hessian vector products (excluding preconditioner builds) : 22

===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian

it= 6 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= True

Residual: r=b-Ax

Preconditioner: M =approx= A^-1

$\|r_0\| = 2.491e+02$

$\|M r_0\| = 8.117e-03$

$(M r_0, r_0) = 6.144e-01$

Iteration : 0 (r, r) = 62055.25627862993

Iteration : 1 (r, r) = 17813.99086948527

Iteration : 2 (r, r) = 3327.8855390116096

Iteration : 3 (r, r) = 2344.986878741371

Iteration : 4 (r, r) = 1070.796146913166

Iteration : 5 (r, r) = 250.5671246548092

Iteration : 6 (r, r) = 107.52551605927837

Iteration : 7 (r, r) = 51.287253784420585

Iteration : 8 (r, r) = 15.591087623569367

Iteration : 9 (r, r) = 7.511362217603848

Iteration : 10 (r, r) = 2.496644433043406

Iteration : 11 (r, r) = 0.9211951070843012

Iteration : 12 (r, r) = 0.4144659637366318

Relative/Absolute residual less than tol

Converged in 12 iterations with final norm 0.6437903103780235

===== Begin Newton CG convergence information =====

=====

Preconditioned inexact Newton-CG with line search

$H_p = -g$

$u \leftarrow u + \alpha * p$

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve $H_p = -g$
 it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve $H_p = -g$

```

...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07      ---- 5.0
e-01
  1   1   1   1   1   T   F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
  2   2   1   1   2   T   F 6.676150e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
  3   5   1   1   5   T   F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
  4   4   1   1   4   T   F 8.249839e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
  5   9   1   1   9   T   F 7.739260e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02
  6  12   1   1  12   T   F 7.735056e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03

converged : False
reason      : unknown reason
cumulative CG iterations : 34
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 34
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= False
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.622e+00
||M r0|| = 1.314e-04
(M r0, r0) = 8.544e-03
Iteration : 0 (r, r) = 2.632324150102312
Iteration : 1 (r, r) = 1.1004878090835133
Iteration : 2 (r, r) = 0.4381865251812377
Iteration : 3 (r, r) = 0.21169953907339312
Iteration : 4 (r, r) = 0.12901175715537366
Iteration : 5 (r, r) = 0.05202813129889891
Iteration : 6 (r, r) = 0.019110488023099646

```

```

Iteration : 7 (r, r) = 0.006849034953224545
Iteration : 8 (r, r) = 0.0027241778745961545
Iteration : 9 (r, r) = 0.0013305864089768202
Iteration : 10 (r, r) = 0.0006622521046680118
Iteration : 11 (r, r) = 0.0002728674586366191
Iteration : 12 (r, r) = 9.594554176196423e-05
Iteration : 13 (r, r) = 3.446989298960136e-05
Iteration : 14 (r, r) = 1.0641595807109882e-05
Iteration : 15 (r, r) = 3.917750475282861e-06
Iteration : 16 (r, r) = 1.2817278787248201e-06
Iteration : 17 (r, r) = 5.022689663570074e-07
Iteration : 18 (r, r) = 2.2573702896450665e-07
Iteration : 19 (r, r) = 7.213822657979337e-08

```

Relative/Absolute residual less than tol

Converged in 19 iterations with final norm 0.00026858560382081795

===== Begin Newton CG convergence information =====

====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

```

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

```

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

||g||L2: l2 norm of gradient

alpha: step size

tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-3.6e+08	2.0e+07	----	5.0
e-01													
1	1	1	1	1	T	F	1.908732e+07	1.8e+07	1.5e+06	-2.6e+07	4.9e+06	1.0e+00	5.0
e-01													
2	2	1	1	2	T	F	6.676150e+06	5.9e+06	7.7e+05	-1.0e+07	2.6e+06	1.0e+00	3.6
e-01													
3	5	1	1	5	T	F	1.532030e+06	1.0e+06	5.3e+05	-1.5e+06	5.7e+05	1.0e+00	1.7
e-01													
4	4	1	1	4	T	F	8.249839e+05	4.2e+05	4.1e+05	-1.0e+05	1.2e+05	1.0e+00	7.9
e-02													
5	9	1	1	9	T	F	7.739260e+05	3.7e+05	4.0e+05	-8.4e+02	7.7e+03	1.0e+00	2.0
e-02													
6	12	1	1	12	T	F	7.735056e+05	3.7e+05	4.0e+05	-3.3e-01	2.5e+02	1.0e+00	3.6

```

e-03
 7 19 1 1 19 F F 7.735055e+05 3.7e+05 4.0e+05 -5.7e-05 1.6e+00 1.0e+00 2.9
e-04

converged : False
reason    : unknown reason
cumulative CG iterations : 53
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====

using Hessian
it= 8 , preconditioner_build_iters= () , num_gn_iter= 7 , using_gauss_newton= Fa
lse

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
se)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha  t
olcg
 0  1  1  1  1  T  F 2.093839e+08 2.1e+08 0.0e+00 -3.6e+08 2.0e+07  ---- 5.0
e-01
 1  1  1  1  1  T  F 1.908732e+07 1.8e+07 1.5e+06 -2.6e+07 4.9e+06 1.0e+00 5.0
e-01
 2  2  1  1  2  T  F 6.676150e+06 5.9e+06 7.7e+05 -1.0e+07 2.6e+06 1.0e+00 3.6
e-01
 3  5  1  1  5  T  F 1.532030e+06 1.0e+06 5.3e+05 -1.5e+06 5.7e+05 1.0e+00 1.7
e-01
 4  4  1  1  4  T  F 8.249839e+05 4.2e+05 4.1e+05 -1.0e+05 1.2e+05 1.0e+00 7.9
e-02
 5  9  1  1  9  T  F 7.739260e+05 3.7e+05 4.0e+05 -8.4e+02 7.7e+03 1.0e+00 2.0
e-02

```

```

 6 12 1 1 12 T F 7.735056e+05 3.7e+05 4.0e+05 -3.3e-01 2.5e+02 1.0e+00 3.6
e-03
 7 19 1 1 19 F F 7.735055e+05 3.7e+05 4.0e+05 -5.7e-05 1.6e+00 1.0e+00 2.9
e-04
 8 0 1 1 0 F F 7.735055e+05 3.7e+05 4.0e+05 ----- 3.6e-02 1.0e+00
-----
converged : True
reason    : Norm of the gradient less than tolerance
cumulative CG iterations : 53
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 53
===== End Newton CG convergence information =====
=====
gamma= 6975.784273393516 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
863.5355313016433
      converged: True
      flag: 'converged'
      function_calls: 9
      iterations: 8
      root: 8.8502000428021
gamma_morozov= 6975.784285975133

```

Plots to make (3):

- Morozov discrepancy log-log plot

In [31]:

```

morozov_gammas = np.array(morozov_gammas)
morozov_discrepancies = np.array(morozov_discrepancies)

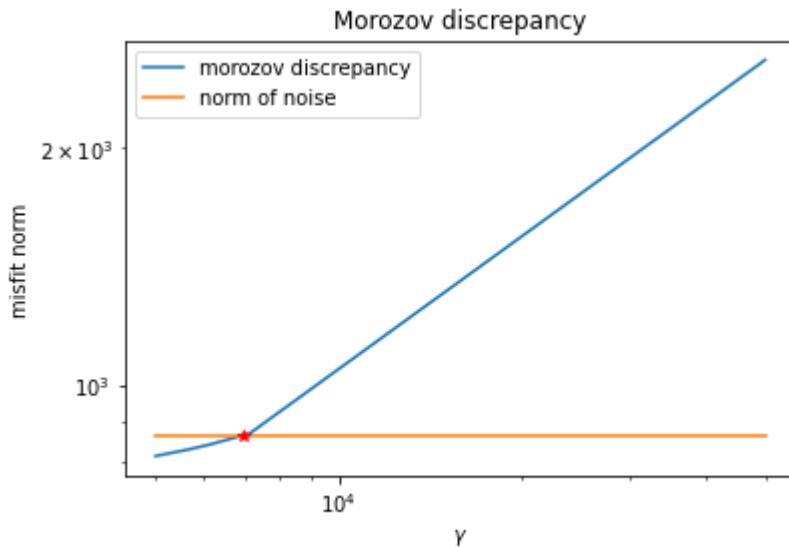
sort_inds = np.argsort(morozov_gammas)
morozov_gammas = morozov_gammas[sort_inds]
morozov_discrepancies = morozov_discrepancies[sort_inds]

np.savetxt(save_dir_str + '/morozov_gammas.txt', morozov_gammas)
np.savetxt(save_dir_str + '/morozov_discrepancies.txt', morozov_discrepancies)
np.savetxt(save_dir_str + '/noise_datanorm.txt', np.array([noise_datanorm]))

plt.figure()
plt.loglog(morozov_gammas, morozov_discrepancies)
plt.loglog(morozov_gammas, noise_datanorm*np.ones(len(morozov_gammas)))
plt.plot(gamma_morozov, noise_datanorm, '*r')
plt.title('Morozov discrepancy')
plt.xlabel(r'$\gamma$')
plt.ylabel('misfit norm')
plt.legend(['morozov discrepancy', 'norm of noise'])

```

Out[31]:



Newton Solve with PCH preconditioning

In [32]:

```
PCH1 = PCHWrapper(do_updates=False)
update_m(m0_constant_value * np.ones(Vh2.dim()))

iter_PCH = list()
m_iter_PCH = list()
def PCH_callback(it, m_Vh2_numpy):
    iter_PCH.append(it)
    m_iter_PCH.append(m_Vh2_numpy)
    m_Vh2 = dl.Function(Vh2)
    m_Vh2.vector()[:] = m_Vh2_numpy
    plt.figure()
    cm = dl.plot(m_Vh2, cmap='gray')
    plt.colorbar(cm)
    plt.title('m_'+str(it))

PCH_convergence_info = newtoncg_ls(get_optimization_variable,
                                    set_optimization_variable,
                                    cost_triple,
                                    gradient,
                                    apply_hessian,
                                    apply_gauss_newton_hessian,
                                    PCH1.build,
                                    update_nothing,
                                    PCH1.solve,
                                    callback=PCH_callback,
                                    preconditioner_build_iters = (3,),
                                    rtol=newton_rtol,
                                    forcing_sequence_power=forcing_sequence_power)

misfit_datanorm = morozov_helper()

using Gauss-Newton Hessian
it= 0 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
```

```

||M r0|| = 1.299e+05
(M r0, r0) = 1.460e+06
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 60792616517657.08
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 7796962.005656888

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07  ---- 5.0
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.905e+06
||M r0|| = 2.779e+04
(M r0, r0) = 4.189e+05
Iteration : 0 (r, r) = 62494263002660.72
Iteration : 1 (r, r) = 25858348443099.72

```

Iteration : 2 (r, r) = 15307513353856.215
 Relative/Absolute residual less than tol
 Converged in 2 iterations with final norm 3912481.7384693585

===== Begin Newton CG convergence information =====

=====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g
 it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g

...

it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

||g||L2: l2 norm of gradient

alpha: step size

tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0 1 1 1 1 T F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	----	5.0
e-01							
1 2 1 1 2 T F	6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
e-01							

converged : False

reason : unknown reason

cumulative CG iterations : 3

cumulative cost evaluations : 2

cumulative gradient evaluations : 2

cumulative Hessian vector products (excluding preconditioner builds) : 3

===== End Newton CG convergence information =====

=====

using Gauss-Newton Hessian

it= 2 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True

Residual: r=b-Ax

Preconditioner: M =approx= A^-1

||r0|| = 3.874e+06

||M r0|| = 1.166e+04

(M r0, r0) = 1.713e+05

Iteration : 0 (r, r) = 15010672486864.512

Iteration : 1 (r, r) = 9835012971501.21

Iteration : 2 (r, r) = 7723910396975.338

```

Iteration : 3 (r, r) = 4596707853392.447
Iteration : 4 (r, r) = 3484908890885.7627
Iteration : 5 (r, r) = 2782124709218.0215
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 1667970.2363105947

```

```
===== Begin Newton CG convergence information =====
```

```
=====
```

```
Preconditioned inexact Newton-CG with line search
```

```
Hp=-g
```

```
u <- u + alpha * p
```

```
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

```
it:      Newton iteration number
```

```
nCG:     number of CG iterations in Newton iteration
```

```
nJ:      number of cost function evaluations in Newton iteration
```

```
nG:      number of gradient evaluations in Newton iteration
```

```
nHp:     number of Hessian-vector products in Newton iteration
```

```
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
```

```
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
```

```
cost:    cost, J = Jd + Jr
```

```
misfit:  misfit cost, Jd
```

```
reg:     regularization cost, Jr
```

```
(g,p):  inner product between gradient, g, and Newton search direction, p
```

```
||g||L2: l2 norm of gradient
```

```
alpha:   step size
```

```
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)
```

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	----	5.0
e-01													
1	2	1	1	2	T	F	6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
e-01													
2	5	1	1	5	T	F	2.180904e+07	2.2e+07	5.1e+04	-3.1e+07	3.9e+06	1.0e+00	4.5
e-01													

```
converged : False
```

```
reason     : unknown reason
```

```
cumulative CG iterations : 8
```

```
cumulative cost evaluations : 3
```

```
cumulative gradient evaluations : 3
```

```
cumulative Hessian vector products (excluding preconditioner builds) : 8
```

```
===== End Newton CG convergence information =====
```

```
=====
```

```
using Gauss-Newton Hessian
```

```
it= 3 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
```

```
building preconditioner
```

```
building PCH preconditioner
```

```
Computing impulse response moments
```

```
getting spatially varying volume
```

```

getting spatially varying mean
getting spatially varying covariance
Preparing sample point batch stuff
Preparing c++ object
Building initial sample point batches

Making row and column cluster treesUsing ProductConvolutionKernelRBFColsOnly!

Making block cluster trees
Building A kernel hmatrix
Making input and output mass matrix hmatrices— building H-matrix ( tol =
Computing A_hmatrix = M_out_hmatrix * A_kernel_hmatrix * M_in_hmatrix
making hmatrix spd
symmetrizing
getting smallest eigenvalue with Lanczos
1e-06 )
    done in 13.44s
    size of H-matrix = 4.42 MB
lambda_min= -76590.67133677886
scaling_at_lambda_min= 0.9878048780487805
scaling_at_zero= 0.5
Setting up operator T = (2*A - (b+a) I) / (b-a)
computing T^(2^k)
computing T^(2^1) = T^(2^0) * T^(2^0)

— H-matrix multiplication C=A*B
done in 32.54s
size of C = 5.27 MB

computing negative spectral projector Pi_minus = I / (I + T^(2^k))
— H-matrix inverse ( rtol = 1e-07 , atol = 1e-12 , overwrite= False )
— H-matrix multiplication C=A*B
[==-
                                done in 23.251529932022095
    size of inverse = 14645377 bytes
computing absolute value projector Pi = I - 2*Pi_minus
                                computing A_plus = Pi * A (5.27 GB)
    done in 33.83s
    size of C = 7.70 MB
)

— H-matrix multiplication C=A*B
done in 36.50s
size of C = 17.55 MB
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is              real valued
format                = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 1.1247e+06
sparsity constant = 20
size of H-matrix     = 4.06 MB
|A|_F                 = 1.1247e+06
|S-A|_2 = 3.13924e-09
matrix has dimension — H-matrix inverse ( rtol = 1416 x 1416
no of non-zeroes      = 9746
matrix is              real valued
format                = symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 5.18561e+06
sparsity constant = 20

```

```

size of H-matrix = 2.29 MB
|A|_F           = 5.18561e+06
|S-A|_2 = 4.66653e-09
1e-06 , atol = 1e-14 , overwrite= False )
done in 14.54411268234253
size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
done in 24.28s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
done in 33.17s
size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
done in 30.85s
size of LU factor = 18.69 MB
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.623e+06
||M r0|| = 3.030e+01
(M r0, r0) = 3.755e+03
Iteration : 0 (r, r) = 2633546763856.491
Iteration : 1 (r, r) = 250580861251.00507
Iteration : 2 (r, r) = 33496680698.47232
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 183020.98431183328

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg

```

```

    0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07      ---- 5.0
e-01
    1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
    2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
    3   2   1   1   2   T   T 6.223754e+06 6.1e+06 1.2e+05 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01

```

```

converged : False
reason     : unknown reason
cumulative CG iterations : 10
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 10
===== End Newton CG convergence information =====
=====
```

```

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 2.050e+05
||M r0|| = 6.264e+00
(M r0, r0) = 6.686e+02
Iteration : 0 (r, r) = 42042735743.235
Iteration : 1 (r, r) = 15671755066.120493
Iteration : 2 (r, r) = 3499198012.00104
Iteration : 3 (r, r) = 1337266702.9177585
Iteration : 4 (r, r) = 446370599.19143844
Iteration : 5 (r, r) = 168805258.69329643
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 12992.507790772974
```

```

===== Begin Newton CG convergence information =====
=====
```

```

Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direct
ion
```

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.
```

```

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwis
e.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
```

```

alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost misfit reg (g,p) ||g||L2 alpha t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07 ---- 5.0
e-01
  1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3   2   1   1   2   T   T 6.223754e+06 6.1e+06 1.2e+05 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
  4   5   1   1   5   T   F 9.138152e+05 4.8e+05 4.3e+05 -2.8e+05 2.1e+05 1.0e+00 1.0
e-01

converged : False
reason   : unknown reason
cumulative CG iterations : 15
cumulative cost evaluations : 5
cumulative gradient evaluations : 5
cumulative Hessian vector products (excluding preconditioner builds) : 15
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 5 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.958e+04
||M r0|| = 5.044e-01
(M r0, r0) = 4.470e+01
Iteration : 0 (r, r) = 383345197.19962656
Iteration : 1 (r, r) = 104864321.5827917
Iteration : 2 (r, r) = 46136303.28401123
Iteration : 3 (r, r) = 39817683.35112052
Iteration : 4 (r, r) = 17026279.5081695
Iteration : 5 (r, r) = 3342774.9987876136
Iteration : 6 (r, r) = 1254356.5052999544
Iteration : 7 (r, r) = 564533.0684810837
Iteration : 8 (r, r) = 209884.7463130094
Relative/Absolute residual less than tol
Converged in 8 iterations with final norm 458.13180015472557

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration

```

nJ: number of cost function evaluations in Newton iteration
nG: number of gradient evaluations in Newton iteration
nHp: number of Hessian-vector products in Newton iteration
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost: cost, $J = J_d + J_r$
misfit: misfit cost, J_d
reg: regularization cost, J_r
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha: step size
tolcg: relative tolerance for $H_p = -g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	----	5.0
e-01													
1	2	1	1	2	T	F	6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
e-01													
2	5	1	1	5	T	F	2.180904e+07	2.2e+07	5.1e+04	-3.1e+07	3.9e+06	1.0e+00	4.5
e-01													
3	2	1	1	2	T	T	6.223754e+06	6.1e+06	1.2e+05	-1.1e+07	1.6e+06	1.0e+00	2.9
e-01													
4	5	1	1	5	T	F	9.138152e+05	4.8e+05	4.3e+05	-2.8e+05	2.1e+05	1.0e+00	1.0
e-01													
5	8	1	1	8	T	F	7.745874e+05	3.7e+05	4.0e+05	-2.2e+03	2.0e+04	1.0e+00	3.2
e-02													

converged : False
reason : unknown reason
cumulative CG iterations : 23
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 23
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M=approx= A^-1
||r0|| = 6.239e+02
||M r0|| = 2.045e-02
(M r0, r0) = 1.620e+00
Iteration : 0 (r, r) = 389309.8150192415
Iteration : 1 (r, r) = 149321.32328557732
Iteration : 2 (r, r) = 48628.06275758954
Iteration : 3 (r, r) = 40624.77118927481
Iteration : 4 (r, r) = 16846.476248559346
Iteration : 5 (r, r) = 4726.28974340554
Iteration : 6 (r, r) = 2304.610238560923
Iteration : 7 (r, r) = 1509.5834879965591
Iteration : 8 (r, r) = 825.8301378265722
Iteration : 9 (r, r) = 146.22076271265655
Iteration : 10 (r, r) = 57.289019345246544
Iteration : 11 (r, r) = 19.72065406458009
Iteration : 12 (r, r) = 6.631543741303982
Relative/Absolute residual less than tol

Converged in 12 iterations with final norm 2.5751783901904703

===== Begin Newton CG convergence information =====

=====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g

it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g

...

it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

||g||L2: l2 norm of gradient

alpha: step size

tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	-----	5.0

e-01

1	2	1	1	2	T	F	6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
---	---	---	---	---	---	---	--------------	---------	---------	----------	---------	---------	-----

e-01

2	5	1	1	5	T	F	2.180904e+07	2.2e+07	5.1e+04	-3.1e+07	3.9e+06	1.0e+00	4.5
---	---	---	---	---	---	---	--------------	---------	---------	----------	---------	---------	-----

e-01

3	2	1	1	2	T	T	6.223754e+06	6.1e+06	1.2e+05	-1.1e+07	1.6e+06	1.0e+00	2.9
---	---	---	---	---	---	---	--------------	---------	---------	----------	---------	---------	-----

e-01

4	5	1	1	5	T	F	9.138152e+05	4.8e+05	4.3e+05	-2.8e+05	2.1e+05	1.0e+00	1.0
---	---	---	---	---	---	---	--------------	---------	---------	----------	---------	---------	-----

e-01

5	8	1	1	8	T	F	7.745874e+05	3.7e+05	4.0e+05	-2.2e+03	2.0e+04	1.0e+00	3.2
---	---	---	---	---	---	---	--------------	---------	---------	----------	---------	---------	-----

e-02

6	12	1	1	12	T	F	7.735069e+05	3.7e+05	4.0e+05	-3.0e+00	6.2e+02	1.0e+00	5.7
---	----	---	---	----	---	---	--------------	---------	---------	----------	---------	---------	-----

e-03

converged : False													
reason : unknown reason													
cumulative CG iterations : 35													
cumulative cost evaluations : 7													
cumulative gradient evaluations : 7													
cumulative Hessian vector products (excluding preconditioner builds) : 35													

===== End Newton CG convergence information =====

=====

using Hessian

it= 7 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton= False

```

Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 5.815e+00
||M r0|| = 4.991e-04
(M r0, r0) = 2.603e-02
  Iteration : 0 (r, r) = 33.818990213008064
  Iteration : 1 (r, r) = 15.463723098182285
  Iteration : 2 (r, r) = 6.28924230001418
  Iteration : 3 (r, r) = 3.8049837307050565
  Iteration : 4 (r, r) = 2.1644727022801864
  Iteration : 5 (r, r) = 0.5541918233703111
  Iteration : 6 (r, r) = 0.1656993563436443
  Iteration : 7 (r, r) = 0.04751182038890159
  Iteration : 8 (r, r) = 0.01601287735337807
  Iteration : 9 (r, r) = 0.006514407673800337
  Iteration : 10 (r, r) = 0.002937606184435253
  Iteration : 11 (r, r) = 0.0009972596591672413
  Iteration : 12 (r, r) = 0.00039270536270223123
  Iteration : 13 (r, r) = 0.00016607290233615143
  Iteration : 14 (r, r) = 5.7395285788918e-05
  Iteration : 15 (r, r) = 1.9187311896355508e-05
  Iteration : 16 (r, r) = 7.5771151067822606e-06
Relative/Absolute residual less than tol
Converged in 16 iterations with final norm 0.002752656009526483

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:  l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07      ---- 5.0
e-01
  1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01

```

```

2 5 1 1 5 T F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
3 2 1 1 2 T T 6.223754e+06 6.1e+06 1.2e+05 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
4 5 1 1 5 T F 9.138152e+05 4.8e+05 4.3e+05 -2.8e+05 2.1e+05 1.0e+00 1.0
e-01
5 8 1 1 8 T F 7.745874e+05 3.7e+05 4.0e+05 -2.2e+03 2.0e+04 1.0e+00 3.2
e-02
6 12 1 1 12 T F 7.735069e+05 3.7e+05 4.0e+05 -3.0e+00 6.2e+02 1.0e+00 5.7
e-03
7 16 1 1 16 F F 7.735055e+05 3.7e+05 4.0e+05 -6.7e-04 5.8e+00 1.0e+00 5.5
e-04

```

```

converged : False
reason    : unknown reason
cumulative CG iterations : 51
cumulative cost evaluations : 8
cumulative gradient evaluations : 8
cumulative Hessian vector products (excluding preconditioner builds) : 51
===== End Newton CG convergence information =====
=====
```

```

using Hessian
it= 8 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
False
```

```

===== Begin Newton CG convergence information =====
=====
```

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

```

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.
```

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

||g||L2: l2 norm of gradient

alpha: step size

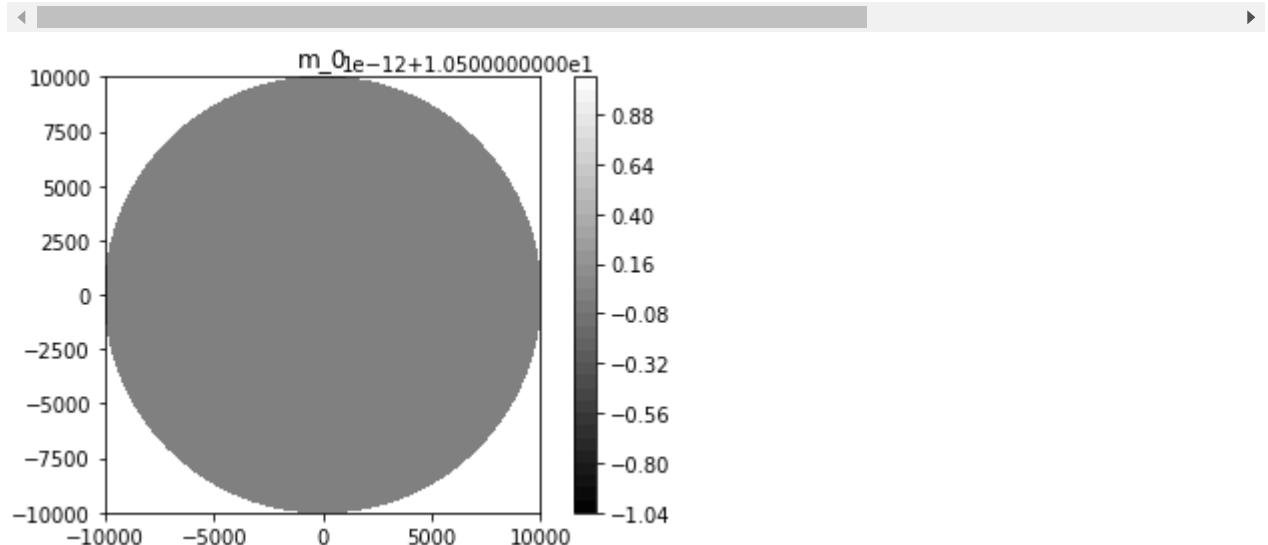
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

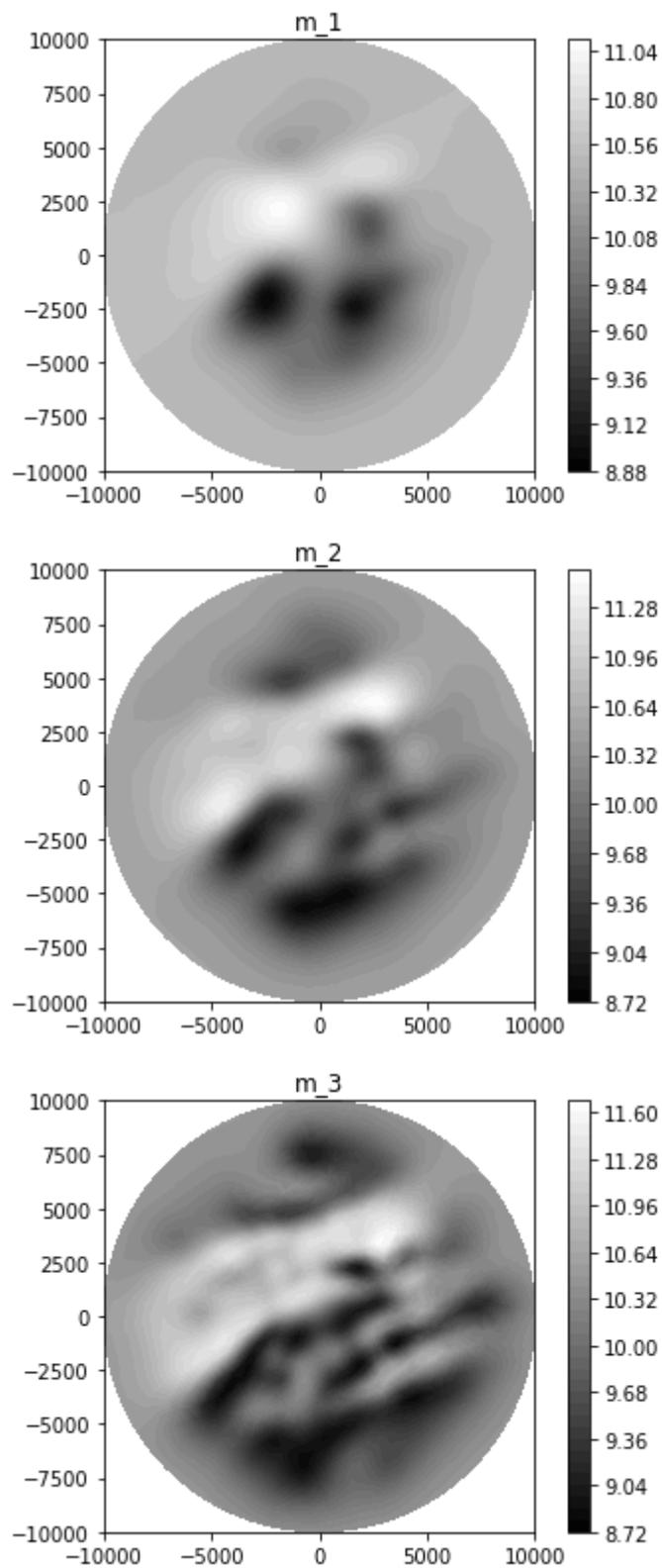
it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0							2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	----	5.0
e-01													
1							6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0

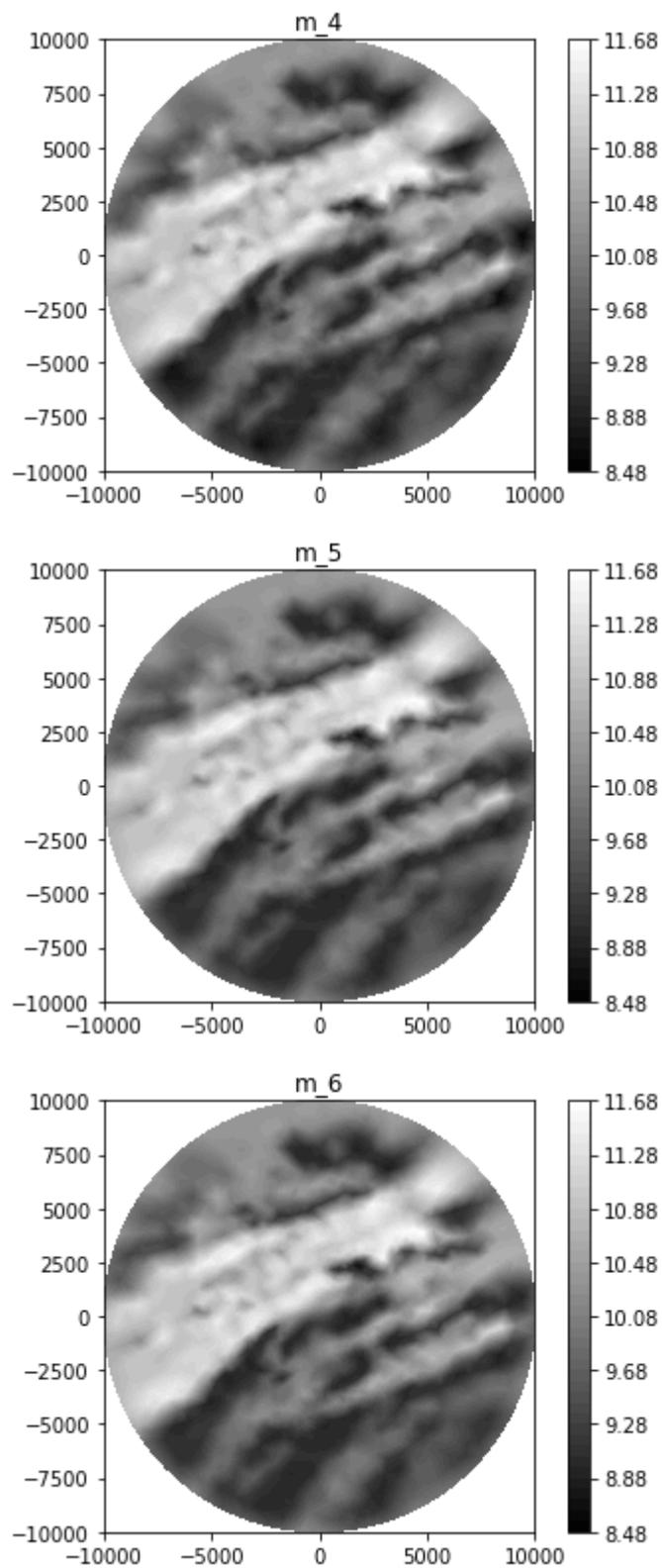
```
e-01
 2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
 3   2   1   1   2   T   T 6.223754e+06 6.1e+06 1.2e+05 -1.1e+07 1.6e+06 1.0e+00 2.9
e-01
 4   5   1   1   5   T   F 9.138152e+05 4.8e+05 4.3e+05 -2.8e+05 2.1e+05 1.0e+00 1.0
e-01
 5   8   1   1   8   T   F 7.745874e+05 3.7e+05 4.0e+05 -2.2e+03 2.0e+04 1.0e+00 3.2
e-02
 6   12  1   1   12  T   F 7.735069e+05 3.7e+05 4.0e+05 -3.0e+00 6.2e+02 1.0e+00 5.7
e-03
 7   16  1   1   16  F   F 7.735055e+05 3.7e+05 4.0e+05 -6.7e-04 5.8e+00 1.0e+00 5.5
e-04
 8   0   1   1   0   F   F 7.735055e+05 3.7e+05 4.0e+05      ---- 4.5e-02 1.0e+00
----
```

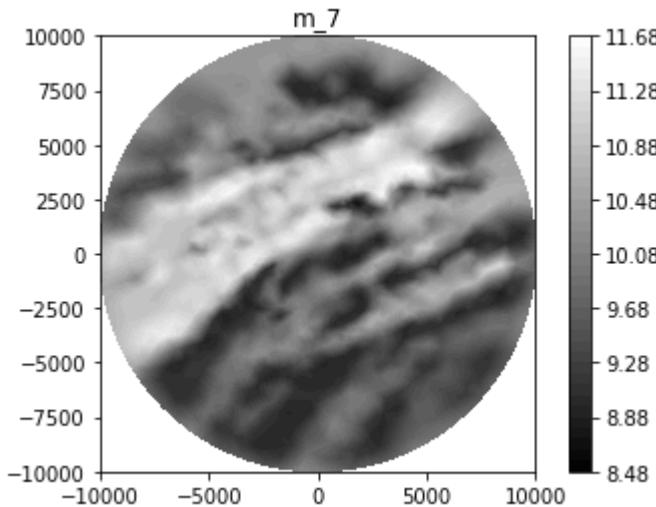
```
converged : True
reason    : Norm of the gradient less than tolerance
cumulative CG iterations : 51
cumulative cost evaluations : 9
cumulative gradient evaluations : 9
cumulative Hessian vector products (excluding preconditioner builds) : 51
===== End Newton CG convergence information =====
=====
```

```
gamma= 6975.784285975133 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
863.5355308077618
```









In [33]:

```
misfit_datanorm = np.sqrt(2.0 * SDAP.misfit())
print('noise_datanorm=', noise_datanorm, ', misfit_datanorm=', misfit_datanorm)
```

```
noise_datanorm= 863.5355316705534 , misfit_datanorm= 863.5355308077618
```

Newton solve with regularization preconditioning

In [34]:

```
update_m(m0_constant_value * np.ones(Vh2.dim()))

iter_REG = list()
m_iter_REG = list()
def REG_callback(it, m_Vh2_numpy):
    iter_REG.append(it)
    m_iter_REG.append(m_Vh2_numpy)
    m_Vh2 = dl.Function(Vh2)
    m_Vh2.vector()[:] = m_Vh2_numpy
    plt.figure()
    cm = dl.plot(m_Vh2, cmap='gray')
    plt.colorbar(cm)
    plt.title('m_'+str(it))

REG_convergence_info = newtoncg_ls(get_optimization_variable,
                                    set_optimization_variable,
                                    cost_triple,
                                    gradient,
                                    apply_hessian,
                                    apply_gauss_newton_hessian,
                                    build_nothing,
                                    update_nothing,
                                    REG.solve_hessian_numpy,
                                    callback=REG_callback,
                                    rtol=newton_rtol,
                                    forcing_sequence_power=forcing_sequence_power)

misfit_datanorm = morozov_helper()

using Gauss-Newton Hessian
it= 0 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
```

```

True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.951e+07
||M r0|| = 1.299e+05
(M r0, r0) = 1.460e+06
Iteration : 0 (r, r) = 380758959655851.75
Iteration : 1 (r, r) = 60792616517657.08
Relative/Absolute residual less than tol
Converged in 1 iterations with final norm 7796962.005656888

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07    ---- 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 1
cumulative cost evaluations : 1
cumulative gradient evaluations : 1
cumulative Hessian vector products (excluding preconditioner builds) : 1
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 1 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.905e+06

```

```

||M r0|| = 2.779e+04
(M r0, r0) = 4.189e+05
Iteration : 0 (r, r) = 62494263002660.72
Iteration : 1 (r, r) = 25858348443099.72
Iteration : 2 (r, r) = 15307513353856.215
Relative/Absolute residual less than tol
Converged in 2 iterations with final norm 3912481.7384693585

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:        Newton iteration number
nCG:       number of CG iterations in Newton iteration
nJ:        number of cost function evaluations in Newton iteration
nG:        number of gradient evaluations in Newton iteration
nHp:       number of Hessian-vector products in Newton iteration
GN:        True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:        True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:      cost, J = Jd + Jr
misfit:    misfit cost, Jd
reg:       regularization cost, Jr
(g,p):    inner product between gradient, g, and Newton search direction, p
||g||L2:   l2 norm of gradient
alpha:     step size
tolcg:    relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
 0  1  1  1   1  T  F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07  ---- 5.0
e-01
 1  2  1  1   2  T  F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01

converged : False
reason     : unknown reason
cumulative CG iterations : 3
cumulative cost evaluations : 2
cumulative gradient evaluations : 2
cumulative Hessian vector products (excluding preconditioner builds) : 3
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 2 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 3.874e+06
||M r0|| = 1.166e+04

```

```

(M r0, r0) = 1.713e+05
Iteration : 0 (r, r) = 15010672486864.512
Iteration : 1 (r, r) = 9835012971501.21
Iteration : 2 (r, r) = 7723910396975.338
Iteration : 3 (r, r) = 4596707853392.447
Iteration : 4 (r, r) = 3484908890885.7627
Iteration : 5 (r, r) = 2782124709218.0215
Relative/Absolute residual less than tol
Converged in 5 iterations with final norm 1667970.2363105947

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:    number of CG iterations in Newton iteration
nJ:     number of cost function evaluations in Newton iteration
nG:     number of gradient evaluations in Newton iteration
nHp:    number of Hessian-vector products in Newton iteration
GN:     True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:     True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:   cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:    regularization cost, Jr
(g,p): inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:  step size
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07    ---- 5.0
e-01
  1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 8
cumulative cost evaluations : 3
cumulative gradient evaluations : 3
cumulative Hessian vector products (excluding preconditioner builds) : 8
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 3 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
True

```

```

building preconditioner
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 1.623e+06
||M r0|| = 4.183e+03
(M r0, r0) = 6.505e+04
  Iteration : 0 (r, r) = 2633546763856.491
  Iteration : 1 (r, r) = 1797914212244.619
  Iteration : 2 (r, r) = 1660712510306.205
  Iteration : 3 (r, r) = 1288347515817.5286
  Iteration : 4 (r, r) = 1078521385205.6877
  Iteration : 5 (r, r) = 1093309960593.429
  Iteration : 6 (r, r) = 867617429843.6278
  Iteration : 7 (r, r) = 621255636446.1287
  Iteration : 8 (r, r) = 441183332097.97125
  Iteration : 9 (r, r) = 386169031113.4435
  Iteration : 10 (r, r) = 342336435763.02625
  Iteration : 11 (r, r) = 300238243125.9479
  Iteration : 12 (r, r) = 290555871850.3236
  Iteration : 13 (r, r) = 227244801699.8081
  Iteration : 14 (r, r) = 204198054022.70337
Relative/Absolute residual less than tol
Converged in 14 iterations with final norm 451882.787924815

===== Begin Newton CG convergence information =====
=====
Preconditioned inexact Newton-CG with line search
Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0      : u=u0          -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1      : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last   : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit:  misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07   ---- 5.0
e-01
  1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5

```

```
e-01
 3 14 1 1 14 T T 6.223754e+06 6.1e+06 1.2e+05 -9.3e+06 1.6e+06 1.0e+00 2.9
e-01

converged : False
reason    : unknown reason
cumulative CG iterations : 22
cumulative cost evaluations : 4
cumulative gradient evaluations : 4
cumulative Hessian vector products (excluding preconditioner builds) : 22
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 4 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.570e+05
||M r0|| = 9.034e+02
(M r0, r0) = 1.480e+04
Iteration : 0 (r, r) = 208814699704.0423
Iteration : 1 (r, r) = 156274225178.69598
Iteration : 2 (r, r) = 151558142671.16113
Iteration : 3 (r, r) = 139091094339.03308
Iteration : 4 (r, r) = 124827891646.63843
Iteration : 5 (r, r) = 118888187106.44522
Iteration : 6 (r, r) = 120794486505.15369
Iteration : 7 (r, r) = 118097898625.76532
Iteration : 8 (r, r) = 111036746788.02646
Iteration : 9 (r, r) = 109337941880.37997
Iteration : 10 (r, r) = 92489603628.72995
Iteration : 11 (r, r) = 93401237923.22665
Iteration : 12 (r, r) = 91017574745.29724
Iteration : 13 (r, r) = 76587142830.50244
Iteration : 14 (r, r) = 77144013743.98473
Iteration : 15 (r, r) = 61378015445.73114
Iteration : 16 (r, r) = 49186435048.63666
Iteration : 17 (r, r) = 40794446122.67066
Iteration : 18 (r, r) = 41165358626.87737
Iteration : 19 (r, r) = 32366270257.165955
Iteration : 20 (r, r) = 24490363937.322876
Iteration : 21 (r, r) = 22375551821.195145
Iteration : 22 (r, r) = 20607845095.096878
Iteration : 23 (r, r) = 19832684577.885567
Iteration : 24 (r, r) = 19033706200.83288
Iteration : 25 (r, r) = 20062166698.41462
Iteration : 26 (r, r) = 18504125522.7228
Iteration : 27 (r, r) = 16579008330.08011
Iteration : 28 (r, r) = 15450347892.541042
Iteration : 29 (r, r) = 14179588778.35804
Iteration : 30 (r, r) = 13444878476.572296
Iteration : 31 (r, r) = 12803163454.573095
Iteration : 32 (r, r) = 13062730201.124844
Iteration : 33 (r, r) = 10815349608.264198
Iteration : 34 (r, r) = 10751956994.9739
Iteration : 35 (r, r) = 9245388494.787855
Iteration : 36 (r, r) = 7906997921.438799
Iteration : 37 (r, r) = 6714593399.253725
Iteration : 38 (r, r) = 6177051751.096422
Iteration : 39 (r, r) = 5296056953.159194
```

```

Iteration : 40 (r, r) = 5233834882.77063
Iteration : 41 (r, r) = 5619340684.244795
Iteration : 42 (r, r) = 4974313791.958543
Iteration : 43 (r, r) = 4536892864.879036
Relative/Absolute residual less than tol
Converged in 43 iterations with final norm 67356.4611962285

```

```
===== Begin Newton CG convergence information =====
```

```
=====
```

```
Preconditioned inexact Newton-CG with line search
```

```
Hp=-g
```

```
u <- u + alpha * p
```

```
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
```

```
it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
```

```
...
```

```
it=last : linesearch u -> J -> g -> Done.
```

```
it: Newton iteration number
```

```
nCG: number of CG iterations in Newton iteration
```

```
nJ: number of cost function evaluations in Newton iteration
```

```
nG: number of gradient evaluations in Newton iteration
```

```
nHp: number of Hessian-vector products in Newton iteration
```

```
GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
```

```
BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
```

```
cost: cost, J = Jd + Jr
```

```
misfit: misfit cost, Jd
```

```
reg: regularization cost, Jr
```

```
(g,p): inner product between gradient, g, and Newton search direction, p
```

```
||g||L2: l2 norm of gradient
```

```
alpha: step size
```

```
tolcg: relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)
```

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	g L2	alpha	t
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	----	5.0
e-01													
1	2	1	1	2	T	F	6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
e-01													
2	5	1	1	5	T	F	2.180904e+07	2.2e+07	5.1e+04	-3.1e+07	3.9e+06	1.0e+00	4.5
e-01													
3	14	1	1	14	T	T	6.223754e+06	6.1e+06	1.2e+05	-9.3e+06	1.6e+06	1.0e+00	2.9
e-01													
4	43	1	1	43	T	F	1.578787e+06	1.3e+06	2.3e+05	-1.6e+06	4.6e+05	1.0e+00	1.5
e-01													

```
converged : False
```

```
reason : unknown reason
```

```
cumulative CG iterations : 65
```

```
cumulative cost evaluations : 5
```

```
cumulative gradient evaluations : 5
```

```
cumulative Hessian vector products (excluding preconditioner builds) : 65
```

```
===== End Newton CG convergence information =====
```

```
=====
```

```
using Gauss-Newton Hessian
```

```
it = 5 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
```

```
True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 7.082e+04
||M r0|| = 1.549e+02
(M r0, r0) = 2.476e+03
Iteration : 0 (r, r) = 5015446977.454357
Iteration : 1 (r, r) = 4226926584.7344627
Iteration : 2 (r, r) = 3363265424.767089
Iteration : 3 (r, r) = 3065205625.257473
Iteration : 4 (r, r) = 2895892680.1961317
Iteration : 5 (r, r) = 2694295995.4240704
Iteration : 6 (r, r) = 2663749270.0654273
Iteration : 7 (r, r) = 2674392231.483264
Iteration : 8 (r, r) = 2474937662.3631306
Iteration : 9 (r, r) = 2471182155.1995716
Iteration : 10 (r, r) = 2439106743.897315
Iteration : 11 (r, r) = 2267162619.562894
Iteration : 12 (r, r) = 2249240345.345931
Iteration : 13 (r, r) = 2201342871.9013405
Iteration : 14 (r, r) = 2075682297.12861
Iteration : 15 (r, r) = 2019005094.7968073
Iteration : 16 (r, r) = 2039068322.4237216
Iteration : 17 (r, r) = 1864430412.6118493
Iteration : 18 (r, r) = 1899908662.222554
Iteration : 19 (r, r) = 1754068401.5958602
Iteration : 20 (r, r) = 1627886921.4528928
Iteration : 21 (r, r) = 1503799574.1301122
Iteration : 22 (r, r) = 1362655895.2547865
Iteration : 23 (r, r) = 1388231238.116437
Iteration : 24 (r, r) = 1248423948.0803251
Iteration : 25 (r, r) = 1265456393.6449885
Iteration : 26 (r, r) = 1216503675.778665
Iteration : 27 (r, r) = 1246855640.0427613
Iteration : 28 (r, r) = 1219735217.0996556
Iteration : 29 (r, r) = 1156914082.6075768
Iteration : 30 (r, r) = 1114715491.1751218
Iteration : 31 (r, r) = 1171956936.1756299
Iteration : 32 (r, r) = 1187055069.3202627
Iteration : 33 (r, r) = 999755685.0054071
Iteration : 34 (r, r) = 809843541.626469
Iteration : 35 (r, r) = 729466418.4460568
Iteration : 36 (r, r) = 658076548.5206707
Iteration : 37 (r, r) = 602093840.7458813
Iteration : 38 (r, r) = 587351585.8145802
Iteration : 39 (r, r) = 520714485.45405227
Iteration : 40 (r, r) = 481512325.3770869
Iteration : 41 (r, r) = 464423675.5660394
Iteration : 42 (r, r) = 460557702.58099496
Iteration : 43 (r, r) = 466164878.4468198
Iteration : 44 (r, r) = 477464220.8458015
Iteration : 45 (r, r) = 505762582.2837423
Iteration : 46 (r, r) = 397796527.03082037
Iteration : 47 (r, r) = 347173334.34753907
Iteration : 48 (r, r) = 296656305.8055004
Iteration : 49 (r, r) = 297043853.5139904
Iteration : 50 (r, r) = 225731696.27324462
Iteration : 51 (r, r) = 211507996.48067746
Iteration : 52 (r, r) = 166827986.70790637
Iteration : 53 (r, r) = 149332495.55657285
Iteration : 54 (r, r) = 140640265.82695824
```

```

Iteration : 55 (r, r) = 129092345.70772202
Iteration : 56 (r, r) = 111639175.09074448
Iteration : 57 (r, r) = 119095368.81640287
Iteration : 58 (r, r) = 108796530.38956907
Iteration : 59 (r, r) = 106722362.2696543
Iteration : 60 (r, r) = 94766493.93004248
Iteration : 61 (r, r) = 90098656.83230785
Iteration : 62 (r, r) = 87562834.06547919
Iteration : 63 (r, r) = 78407669.47752306
Iteration : 64 (r, r) = 78513738.26139826
Iteration : 65 (r, r) = 71323276.4488349
Iteration : 66 (r, r) = 67457491.63288079
Iteration : 67 (r, r) = 69746637.77956004
Iteration : 68 (r, r) = 62105544.51546043
Iteration : 69 (r, r) = 66286379.850912064
Iteration : 70 (r, r) = 54802885.828488916
Iteration : 71 (r, r) = 65354458.55278109
Iteration : 72 (r, r) = 54401053.31802486
Iteration : 73 (r, r) = 59142553.55050546
Iteration : 74 (r, r) = 56409531.57357912
Iteration : 75 (r, r) = 57345472.46728028
Iteration : 76 (r, r) = 50227049.91947527
Iteration : 77 (r, r) = 45988631.34193316
Iteration : 78 (r, r) = 47356475.87091111
Iteration : 79 (r, r) = 39119412.330537036
Iteration : 80 (r, r) = 36366548.49069657
Iteration : 81 (r, r) = 30454649.533442654
Iteration : 82 (r, r) = 30886333.70319294
Iteration : 83 (r, r) = 26557315.929344974
Iteration : 84 (r, r) = 23989731.761429753
Iteration : 85 (r, r) = 24191600.979336746
Iteration : 86 (r, r) = 21760403.168978468
Iteration : 87 (r, r) = 22696557.77637533
Iteration : 88 (r, r) = 19470611.01371662
Iteration : 89 (r, r) = 15711254.562275952

```

Relative/Absolute residual less than tol

Converged in 89 iterations with final norm 3963.742494445868

===== Begin Newton CG convergence information =====

====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g
 it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g

...

it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

```

reg:      regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:    step size
tolcg:   relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit    reg  (g,p) ||g||L2  alpha  t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07  ---- 5.0
e-01
  1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3  14   1   1  14   T   T 6.223754e+06 6.1e+06 1.2e+05 -9.3e+06 1.6e+06 1.0e+00 2.9
e-01
  4  43   1   1  43   T   F 1.578787e+06 1.3e+06 2.3e+05 -1.6e+06 4.6e+05 1.0e+00 1.5
e-01
  5  89   1   1  89   T   F 7.995442e+05 4.3e+05 3.7e+05 -5.2e+04 7.1e+04 1.0e+00 6.0
e-02

converged : False
reason     : unknown reason
cumulative CG iterations : 154
cumulative cost evaluations : 6
cumulative gradient evaluations : 6
cumulative Hessian vector products (excluding preconditioner builds) : 154
===== End Newton CG convergence information =====
=====

using Gauss-Newton Hessian
it= 6 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=True
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 4.116e+03
||M r0|| = 7.644e+00
(M r0, r0) = 1.156e+02
Iteration : 0 (r, r) = 16939358.21579752
Iteration : 1 (r, r) = 15829940.842864025
Iteration : 2 (r, r) = 13676674.18744998
Iteration : 3 (r, r) = 13000319.677204274
Iteration : 4 (r, r) = 12839903.085058603
Iteration : 5 (r, r) = 11252931.106234163
Iteration : 6 (r, r) = 11117532.353566494
Iteration : 7 (r, r) = 10454889.809724286
Iteration : 8 (r, r) = 10473481.648212474
Iteration : 9 (r, r) = 9995063.508553853
Iteration : 10 (r, r) = 10032096.937647901
Iteration : 11 (r, r) = 9773935.1979649
Iteration : 12 (r, r) = 10054966.654117588
Iteration : 13 (r, r) = 9698673.34881819
Iteration : 14 (r, r) = 9245175.130086863
Iteration : 15 (r, r) = 8425372.838980898
Iteration : 16 (r, r) = 8095330.444068493
Iteration : 17 (r, r) = 7218567.876135841
Iteration : 18 (r, r) = 6627809.069332445
Iteration : 19 (r, r) = 6419969.876993366
Iteration : 20 (r, r) = 5726033.871260938
Iteration : 21 (r, r) = 5370349.602071915

```

Iteration : 22 (r, r) = 5336818.750085354
Iteration : 23 (r, r) = 5112306.267622681
Iteration : 24 (r, r) = 5249001.404878503
Iteration : 25 (r, r) = 5100659.549067544
Iteration : 26 (r, r) = 4951474.685012238
Iteration : 27 (r, r) = 4740931.757715403
Iteration : 28 (r, r) = 4084323.088254988
Iteration : 29 (r, r) = 4120431.0633110167
Iteration : 30 (r, r) = 3925581.033911786
Iteration : 31 (r, r) = 3619150.5510794306
Iteration : 32 (r, r) = 3893175.7999986187
Iteration : 33 (r, r) = 3938525.778453321
Iteration : 34 (r, r) = 3632118.344556088
Iteration : 35 (r, r) = 3611091.2217316125
Iteration : 36 (r, r) = 3581500.8359455946
Iteration : 37 (r, r) = 3134781.4435195243
Iteration : 38 (r, r) = 2945147.370118011
Iteration : 39 (r, r) = 2768632.5140945353
Iteration : 40 (r, r) = 2643943.6410198957
Iteration : 41 (r, r) = 2509313.4685574668
Iteration : 42 (r, r) = 2525362.050482229
Iteration : 43 (r, r) = 2503614.002094605
Iteration : 44 (r, r) = 2433353.3767861007
Iteration : 45 (r, r) = 2279107.903945623
Iteration : 46 (r, r) = 2436987.664032352
Iteration : 47 (r, r) = 2057757.6094985926
Iteration : 48 (r, r) = 1867418.2079785536
Iteration : 49 (r, r) = 1602104.3682036689
Iteration : 50 (r, r) = 1401867.1044460777
Iteration : 51 (r, r) = 1583244.829775154
Iteration : 52 (r, r) = 1211240.1496423692
Iteration : 53 (r, r) = 1099416.808403374
Iteration : 54 (r, r) = 1144208.1836736002
Iteration : 55 (r, r) = 959710.3778465224
Iteration : 56 (r, r) = 929986.8833463367
Iteration : 57 (r, r) = 956057.8332374836
Iteration : 58 (r, r) = 848423.3046664663
Iteration : 59 (r, r) = 852778.2829900356
Iteration : 60 (r, r) = 803417.9227235925
Iteration : 61 (r, r) = 786807.3034546932
Iteration : 62 (r, r) = 688890.398518729
Iteration : 63 (r, r) = 640988.2662193326
Iteration : 64 (r, r) = 572724.8220247928
Iteration : 65 (r, r) = 522808.34792641096
Iteration : 66 (r, r) = 462477.1346515586
Iteration : 67 (r, r) = 449178.817166253
Iteration : 68 (r, r) = 402003.2518144209
Iteration : 69 (r, r) = 363891.0248270657
Iteration : 70 (r, r) = 386721.07399884146
Iteration : 71 (r, r) = 394908.4263579035
Iteration : 72 (r, r) = 320555.25108559534
Iteration : 73 (r, r) = 289994.12139775674
Iteration : 74 (r, r) = 298829.38327594043
Iteration : 75 (r, r) = 295065.2895523781
Iteration : 76 (r, r) = 272769.4997038597
Iteration : 77 (r, r) = 239906.0208355223
Iteration : 78 (r, r) = 244573.04901723412
Iteration : 79 (r, r) = 226412.17988788214
Iteration : 80 (r, r) = 224791.61556675954
Iteration : 81 (r, r) = 190442.29768792976
Iteration : 82 (r, r) = 223710.99710221085

```

Iteration : 83 (r, r) = 188888.5667581929
Iteration : 84 (r, r) = 171893.79806178517
Iteration : 85 (r, r) = 156179.34690351546
Iteration : 86 (r, r) = 156974.41108843716
Iteration : 87 (r, r) = 145485.56109250695
Iteration : 88 (r, r) = 146058.1677343908
Iteration : 89 (r, r) = 166649.29226406774
Iteration : 90 (r, r) = 165773.34212792426
Iteration : 91 (r, r) = 167302.90164516587
Iteration : 92 (r, r) = 132887.5852534274
Iteration : 93 (r, r) = 117611.97769440032
Iteration : 94 (r, r) = 97165.91166148696
Iteration : 95 (r, r) = 101226.10529205346
Iteration : 96 (r, r) = 102333.03931087881
Iteration : 97 (r, r) = 68459.04460998529
Iteration : 98 (r, r) = 67136.11753525406
Iteration : 99 (r, r) = 53694.62540588234
Iteration : 100 (r, r) = 52037.305645079075
Iteration : 101 (r, r) = 51502.65928632441
Iteration : 102 (r, r) = 40210.428838687556
Iteration : 103 (r, r) = 43180.85507375995
Iteration : 104 (r, r) = 34839.62640147778
Iteration : 105 (r, r) = 34799.50712406801
Iteration : 106 (r, r) = 32060.363863718914
Iteration : 107 (r, r) = 26668.910943214672
Iteration : 108 (r, r) = 28209.541591361587
Iteration : 109 (r, r) = 24351.569981106713
Iteration : 110 (r, r) = 24618.85152375905
Iteration : 111 (r, r) = 22574.1255187585
Iteration : 112 (r, r) = 24083.233419933353
Iteration : 113 (r, r) = 19471.090558898683
Iteration : 114 (r, r) = 17581.983650198323
Iteration : 115 (r, r) = 14966.4278973316
Iteration : 116 (r, r) = 16176.492725193664
Iteration : 117 (r, r) = 14274.473062932364
Iteration : 118 (r, r) = 13842.43236500987
Iteration : 119 (r, r) = 11565.902571608436
Iteration : 120 (r, r) = 10105.482899867682
Iteration : 121 (r, r) = 9851.00982699451
Iteration : 122 (r, r) = 9669.446252100868
Iteration : 123 (r, r) = 8491.918126617493
Iteration : 124 (r, r) = 9159.148313408537
Iteration : 125 (r, r) = 7409.885204931529
Iteration : 126 (r, r) = 6873.122725768691
Iteration : 127 (r, r) = 6788.865461705542
Iteration : 128 (r, r) = 5760.6773020573555
Iteration : 129 (r, r) = 5991.673137903212
Iteration : 130 (r, r) = 5259.797881030383
Iteration : 131 (r, r) = 4750.881974249822
Iteration : 132 (r, r) = 4497.623487228953
Iteration : 133 (r, r) = 4228.438088141882
Iteration : 134 (r, r) = 4440.442729312833
Iteration : 135 (r, r) = 3699.046457289701
Iteration : 136 (r, r) = 4055.216109672579
Iteration : 137 (r, r) = 3449.8565814041594

```

Relative/Absolute residual less than tol

Converged in 137 iterations with final norm 58.735479749502

===== Begin Newton CG convergence information =====

====

Preconditioned inexact Newton-CG with line search

```

Hp=-g
u <- u + alpha * p
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0    : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1    : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
...
it=last : linesearch u -> J -> g -> Done.

it:      Newton iteration number
nCG:     number of CG iterations in Newton iteration
nJ:      number of cost function evaluations in Newton iteration
nG:      number of gradient evaluations in Newton iteration
nHp:     number of Hessian-vector products in Newton iteration
GN:      True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
BP:      True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
cost:    cost, J = Jd + Jr
misfit: misfit cost, Jd
reg:     regularization cost, Jr
(g,p):   inner product between gradient, g, and Newton search direction, p
||g||L2: l2 norm of gradient
alpha:   step size
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrease)

it nCG nJ nG nHp GN BP          cost  misfit      reg    (g,p) ||g||L2  alpha   t
olcg
  0   1   1   1   1   T   F 2.093839e+08 2.1e+08 0.0e+00 -2.9e+08 2.0e+07      ---- 5.0
e-01
  1   2   1   1   2   T   F 6.081021e+07 6.1e+07 2.0e+04 -7.7e+07 7.9e+06 1.0e+00 5.0
e-01
  2   5   1   1   5   T   F 2.180904e+07 2.2e+07 5.1e+04 -3.1e+07 3.9e+06 1.0e+00 4.5
e-01
  3  14   1   1  14   T   T 6.223754e+06 6.1e+06 1.2e+05 -9.3e+06 1.6e+06 1.0e+00 2.9
e-01
  4  43   1   1  43   T   F 1.578787e+06 1.3e+06 2.3e+05 -1.6e+06 4.6e+05 1.0e+00 1.5
e-01
  5  89   1   1  89   T   F 7.995442e+05 4.3e+05 3.7e+05 -5.2e+04 7.1e+04 1.0e+00 6.0
e-02
  6 137   1   1 137   T   F 7.736097e+05 3.7e+05 4.0e+05 -2.1e+02 4.1e+03 1.0e+00 1.5
e-02

converged : False
reason      : unknown reason
cumulative CG iterations : 291
cumulative cost evaluations : 7
cumulative gradient evaluations : 7
cumulative Hessian vector products (excluding preconditioner builds) : 291
===== End Newton CG convergence information =====
=====

using Hessian
it= 7 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=
False
Residual: r=b-Ax
Preconditioner: M =approx= A^-1
||r0|| = 6.841e+01
||M r0|| = 1.358e-01
(M r0, r0) = 2.103e+00

```

Iteration : 0 (r, r) = 4680.392646908169
Iteration : 1 (r, r) = 4125.295051961384
Iteration : 2 (r, r) = 3570.7867989924634
Iteration : 3 (r, r) = 3771.976830214595
Iteration : 4 (r, r) = 3671.394206207255
Iteration : 5 (r, r) = 3312.7577119459775
Iteration : 6 (r, r) = 3721.5064242710087
Iteration : 7 (r, r) = 3213.1669008171484
Iteration : 8 (r, r) = 3309.287436353742
Iteration : 9 (r, r) = 3420.335177030822
Iteration : 10 (r, r) = 3247.873061773187
Iteration : 11 (r, r) = 3155.130485673222
Iteration : 12 (r, r) = 3286.953993505009
Iteration : 13 (r, r) = 3084.6154189378385
Iteration : 14 (r, r) = 2821.258128727842
Iteration : 15 (r, r) = 2804.1359708451337
Iteration : 16 (r, r) = 2651.600236407752
Iteration : 17 (r, r) = 2575.1569320326835
Iteration : 18 (r, r) = 2421.8591473017564
Iteration : 19 (r, r) = 2379.252628879637
Iteration : 20 (r, r) = 2221.3606985140564
Iteration : 21 (r, r) = 2053.137391125238
Iteration : 22 (r, r) = 2101.5220486625676
Iteration : 23 (r, r) = 2132.2451257003145
Iteration : 24 (r, r) = 2065.1842201354225
Iteration : 25 (r, r) = 1974.6594957525162
Iteration : 26 (r, r) = 1836.1499633367102
Iteration : 27 (r, r) = 1749.5913182016552
Iteration : 28 (r, r) = 1686.6753582815124
Iteration : 29 (r, r) = 1652.822124245469
Iteration : 30 (r, r) = 1477.0821745084106
Iteration : 31 (r, r) = 1475.4020794142273
Iteration : 32 (r, r) = 1482.3647251149264
Iteration : 33 (r, r) = 1319.051816323662
Iteration : 34 (r, r) = 1231.384509488952
Iteration : 35 (r, r) = 1152.715529036428
Iteration : 36 (r, r) = 1036.5747273685474
Iteration : 37 (r, r) = 1003.747247756638
Iteration : 38 (r, r) = 951.5576342408037
Iteration : 39 (r, r) = 909.3720215978352
Iteration : 40 (r, r) = 855.3747459810445
Iteration : 41 (r, r) = 910.453435778993
Iteration : 42 (r, r) = 855.3709843228849
Iteration : 43 (r, r) = 813.1830992632001
Iteration : 44 (r, r) = 823.1334508962059
Iteration : 45 (r, r) = 755.3497859962672
Iteration : 46 (r, r) = 702.6525528831562
Iteration : 47 (r, r) = 709.9724640457166
Iteration : 48 (r, r) = 677.5598368649179
Iteration : 49 (r, r) = 601.4270860250836
Iteration : 50 (r, r) = 642.7149247447705
Iteration : 51 (r, r) = 553.714303259302
Iteration : 52 (r, r) = 506.22036055772423
Iteration : 53 (r, r) = 523.7209960183188
Iteration : 54 (r, r) = 433.1275655734473
Iteration : 55 (r, r) = 364.6176476482824
Iteration : 56 (r, r) = 354.98926303217485
Iteration : 57 (r, r) = 369.6713523861059
Iteration : 58 (r, r) = 300.9088481998485
Iteration : 59 (r, r) = 273.1113300877132
Iteration : 60 (r, r) = 281.5484238499139

Iteration : 61 (r, r) = 264.8181460079386
Iteration : 62 (r, r) = 285.36321038634503
Iteration : 63 (r, r) = 234.0417146377423
Iteration : 64 (r, r) = 203.0694059439502
Iteration : 65 (r, r) = 207.55719934638637
Iteration : 66 (r, r) = 170.48283365546456
Iteration : 67 (r, r) = 152.65372043551918
Iteration : 68 (r, r) = 149.14819114917975
Iteration : 69 (r, r) = 129.8359957193319
Iteration : 70 (r, r) = 128.11965976720126
Iteration : 71 (r, r) = 116.81292916576874
Iteration : 72 (r, r) = 104.76204432428133
Iteration : 73 (r, r) = 104.42263356681264
Iteration : 74 (r, r) = 99.69168960650548
Iteration : 75 (r, r) = 106.99407276562034
Iteration : 76 (r, r) = 84.87805819638604
Iteration : 77 (r, r) = 78.99305963043359
Iteration : 78 (r, r) = 77.42621601533313
Iteration : 79 (r, r) = 81.72445706828888
Iteration : 80 (r, r) = 79.95118184009556
Iteration : 81 (r, r) = 83.08786826210033
Iteration : 82 (r, r) = 69.66739338385287
Iteration : 83 (r, r) = 67.37027089055397
Iteration : 84 (r, r) = 63.52627698908434
Iteration : 85 (r, r) = 59.51788386615817
Iteration : 86 (r, r) = 54.27958679415797
Iteration : 87 (r, r) = 53.975569114629884
Iteration : 88 (r, r) = 51.38871441161929
Iteration : 89 (r, r) = 43.458274808963694
Iteration : 90 (r, r) = 43.26307153252954
Iteration : 91 (r, r) = 41.158981676196106
Iteration : 92 (r, r) = 42.04407302511191
Iteration : 93 (r, r) = 38.07955602885774
Iteration : 94 (r, r) = 34.49310133442029
Iteration : 95 (r, r) = 35.66331608526443
Iteration : 96 (r, r) = 36.18038852104269
Iteration : 97 (r, r) = 29.552178153534854
Iteration : 98 (r, r) = 34.00489136798697
Iteration : 99 (r, r) = 27.055125492969132
Iteration : 100 (r, r) = 23.990591290819037
Iteration : 101 (r, r) = 23.8595822107147
Iteration : 102 (r, r) = 18.12485894144455
Iteration : 103 (r, r) = 19.22384510369739
Iteration : 104 (r, r) = 16.519220985581242
Iteration : 105 (r, r) = 16.62171487780605
Iteration : 106 (r, r) = 14.473290521789258
Iteration : 107 (r, r) = 14.674954409377849
Iteration : 108 (r, r) = 12.514701746631474
Iteration : 109 (r, r) = 13.70061409240968
Iteration : 110 (r, r) = 11.6703426684449
Iteration : 111 (r, r) = 11.211845731277151
Iteration : 112 (r, r) = 11.774597389386162
Iteration : 113 (r, r) = 10.384368278004377
Iteration : 114 (r, r) = 10.63791774735525
Iteration : 115 (r, r) = 9.87316965739577
Iteration : 116 (r, r) = 8.918127340856994
Iteration : 117 (r, r) = 10.683883609570572
Iteration : 118 (r, r) = 8.157487133311948
Iteration : 119 (r, r) = 7.934014640171668
Iteration : 120 (r, r) = 6.971172330559067
Iteration : 121 (r, r) = 6.5091676141562385

Iteration : 122 (r, r) = 6.27194490888004
Iteration : 123 (r, r) = 5.270275741557111
Iteration : 124 (r, r) = 4.835730272439543
Iteration : 125 (r, r) = 5.086948679305122
Iteration : 126 (r, r) = 4.899524752774311
Iteration : 127 (r, r) = 4.544025022882121
Iteration : 128 (r, r) = 3.8372105499989053
Iteration : 129 (r, r) = 3.9869242522528827
Iteration : 130 (r, r) = 3.118803422519643
Iteration : 131 (r, r) = 2.8485665986646778
Iteration : 132 (r, r) = 2.785138110628126
Iteration : 133 (r, r) = 2.673457280689694
Iteration : 134 (r, r) = 2.5496742076636583
Iteration : 135 (r, r) = 2.383969666569546
Iteration : 136 (r, r) = 2.187770115319438
Iteration : 137 (r, r) = 2.6327844581605735
Iteration : 138 (r, r) = 2.244977017001406
Iteration : 139 (r, r) = 1.8441373815371702
Iteration : 140 (r, r) = 1.8059530694850436
Iteration : 141 (r, r) = 1.7441619001221698
Iteration : 142 (r, r) = 1.5368610930642599
Iteration : 143 (r, r) = 1.6158540668184316
Iteration : 144 (r, r) = 1.223951089857826
Iteration : 145 (r, r) = 1.3467021454426398
Iteration : 146 (r, r) = 1.0617517053766816
Iteration : 147 (r, r) = 1.0239121633656532
Iteration : 148 (r, r) = 1.0084510086331964
Iteration : 149 (r, r) = 0.9635221060606705
Iteration : 150 (r, r) = 0.893076935501322
Iteration : 151 (r, r) = 0.8559293176223289
Iteration : 152 (r, r) = 0.7353157010827055
Iteration : 153 (r, r) = 0.8267628835571275
Iteration : 154 (r, r) = 0.6507583938446451
Iteration : 155 (r, r) = 0.6463162311819997
Iteration : 156 (r, r) = 0.5421571528223947
Iteration : 157 (r, r) = 0.49750459010952125
Iteration : 158 (r, r) = 0.5279000398746043
Iteration : 159 (r, r) = 0.4754291369378048
Iteration : 160 (r, r) = 0.47039733637700587
Iteration : 161 (r, r) = 0.3874828851864443
Iteration : 162 (r, r) = 0.36691537469590774
Iteration : 163 (r, r) = 0.4123569000570968
Iteration : 164 (r, r) = 0.38441890363422004
Iteration : 165 (r, r) = 0.3333579298331476
Iteration : 166 (r, r) = 0.31156491571040745
Iteration : 167 (r, r) = 0.27219968812199025
Iteration : 168 (r, r) = 0.27578480208396117
Iteration : 169 (r, r) = 0.2398043637198655
Iteration : 170 (r, r) = 0.23456731584957974
Iteration : 171 (r, r) = 0.22223224772953254
Iteration : 172 (r, r) = 0.23865907114845222
Iteration : 173 (r, r) = 0.1983149049734706
Iteration : 174 (r, r) = 0.23990648985795715
Iteration : 175 (r, r) = 0.1819098298466856
Iteration : 176 (r, r) = 0.1701360872768624
Iteration : 177 (r, r) = 0.1567644614047008
Iteration : 178 (r, r) = 0.14819074540635124
Iteration : 179 (r, r) = 0.14118284746866672
Iteration : 180 (r, r) = 0.150794916653695
Iteration : 181 (r, r) = 0.13427128403133148
Iteration : 182 (r, r) = 0.11579318556919715

```

Iteration : 183 (r, r) = 0.10650846048683599
Iteration : 184 (r, r) = 0.09956408986540763
Iteration : 185 (r, r) = 0.10673307335008737
Iteration : 186 (r, r) = 0.08935995206142411
Iteration : 187 (r, r) = 0.08466192639984654
Iteration : 188 (r, r) = 0.08508403091309337
Iteration : 189 (r, r) = 0.06717343504233254
Iteration : 190 (r, r) = 0.0710844153338924
Iteration : 191 (r, r) = 0.06838014894964176
Iteration : 192 (r, r) = 0.053089076681112735
Iteration : 193 (r, r) = 0.05794948996029318
Iteration : 194 (r, r) = 0.05335007518443348
Iteration : 195 (r, r) = 0.05277685192086817
Iteration : 196 (r, r) = 0.04492426513354285
Iteration : 197 (r, r) = 0.04585274923738247
Iteration : 198 (r, r) = 0.03586349698197262
Iteration : 199 (r, r) = 0.04099082911719869
Iteration : 200 (r, r) = 0.035647617821953075
Iteration : 201 (r, r) = 0.03625441777487742
Iteration : 202 (r, r) = 0.03276052532271956
Iteration : 203 (r, r) = 0.03234473137515734
Iteration : 204 (r, r) = 0.028902904819872836
Iteration : 205 (r, r) = 0.023775574665537418
Iteration : 206 (r, r) = 0.02604100631198865
Iteration : 207 (r, r) = 0.021139723632776185
Iteration : 208 (r, r) = 0.020506606500749473
Iteration : 209 (r, r) = 0.019229319034283807
Iteration : 210 (r, r) = 0.019362233407578807
Iteration : 211 (r, r) = 0.017265726108519425
Iteration : 212 (r, r) = 0.014864424413508052
Relative/Absolute residual less than tol
Converged in 212 iterations with final norm 0.12191974579004031

```

```
===== Begin Newton CG convergence information =====
```

```
====
```

```
Preconditioned inexact Newton-CG with line search
```

```
Hp=-g
```

```
u <- u + alpha * p
```

```
u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction
```

```
it=0 : u=u0      -> J -> g -> build precond (optional) -> cgssolve Hp=-g
it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g
```

```
...
```

```
it=last : linesearch u -> J -> g -> Done.
```

```
it:      Newton iteration number
```

```
nCG:    number of CG iterations in Newton iteration
```

```
nJ:    number of cost function evaluations in Newton iteration
```

```
nG:    number of gradient evaluations in Newton iteration
```

```
nHp:    number of Hessian-vector products in Newton iteration
```

```
GN:    True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used
```

```
BP:    True (T) if we built or rebuilt the preconditioner, False (F) otherwise.
```

```
cost:   cost, J = Jd + Jr
```

```
misfit: misfit cost, Jd
```

```
reg:    regularization cost, Jr
```

```
(g,p): inner product between gradient, g, and Newton search direction, p
```

```
||g||L2: l2 norm of gradient
```

```
alpha:   step size
```

```
tolcg:  relative tolerance for Hp=-g CG solve (unpreconditioned residual decrea
```

se)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t	
0	1	1	1	1	T	F	2.093839e+08	2.1e+08	0.0e+00	-2.9e+08	2.0e+07	----	5.0	
e-01	1	2	1	1	2	T	F	6.081021e+07	6.1e+07	2.0e+04	-7.7e+07	7.9e+06	1.0e+00	5.0
e-01	2	5	1	1	5	T	F	2.180904e+07	2.2e+07	5.1e+04	-3.1e+07	3.9e+06	1.0e+00	4.5
e-01	3	14	1	1	14	T	T	6.223754e+06	6.1e+06	1.2e+05	-9.3e+06	1.6e+06	1.0e+00	2.9
e-01	4	43	1	1	43	T	F	1.578787e+06	1.3e+06	2.3e+05	-1.6e+06	4.6e+05	1.0e+00	1.5
e-01	5	89	1	1	89	T	F	7.995442e+05	4.3e+05	3.7e+05	-5.2e+04	7.1e+04	1.0e+00	6.0
e-02	6	137	1	1	137	T	F	7.736097e+05	3.7e+05	4.0e+05	-2.1e+02	4.1e+03	1.0e+00	1.5
e-02	7	212	1	1	212	F	F	7.735055e+05	3.7e+05	4.0e+05	-8.1e-02	6.8e+01	1.0e+00	1.9
e-03														

converged : False

reason : unknown reason

cumulative CG iterations : 503

cumulative cost evaluations : 8

cumulative gradient evaluations : 8

cumulative Hessian vector products (excluding preconditioner builds) : 503

===== End Newton CG convergence information ======
=====

using Hessian

it= 8 , preconditioner_build_iters= (3,) , num_gn_iter= 7 , using_gauss_newton=False

===== Begin Newton CG convergence information ======
=====

Preconditioned inexact Newton-CG with line search

Hp=-g

u <- u + alpha * p

u: parameter, J: cost, g: gradient, H: Hessian, alpha=step size, p=search direction

it=0 : u=u0 -> J -> g -> build precond (optional) -> cgssolve Hp=-g

it=1 : linesearch u -> J -> g -> build precond (optional) -> cgssolve Hp=-g

...

it=last : linesearch u -> J -> g -> Done.

it: Newton iteration number

nCG: number of CG iterations in Newton iteration

nJ: number of cost function evaluations in Newton iteration

nG: number of gradient evaluations in Newton iteration

nHp: number of Hessian-vector products in Newton iteration

GN: True (T) if Gauss-Newton Hessian is used, False (F) if Hessian is used

BP: True (T) if we built or rebuilt the preconditioner, False (F) otherwise.

cost: cost, J = Jd + Jr

misfit: misfit cost, Jd

reg: regularization cost, Jr

(g,p): inner product between gradient, g, and Newton search direction, p

 $\|g\|_{L2}$: l2 norm of gradient

alpha: step size

tolcg: relative tolerance for $H_p = g$ CG solve (unpreconditioned residual decrease)

it	nCG	nJ	nG	nHp	GN	BP	cost	misfit	reg	(g,p)	$\ g\ _{L2}$	alpha	t
0	1	1	1	1	T	F	$2.093839e+08$	$2.1e+08$	$0.0e+00$	$-2.9e+08$	$2.0e+07$	----	5.0
e-01	1	2	1	1	2	T	$6.081021e+07$	$6.1e+07$	$2.0e+04$	$-7.7e+07$	$7.9e+06$	$1.0e+00$	5.0
e-01	2	5	1	1	5	T	$2.180904e+07$	$2.2e+07$	$5.1e+04$	$-3.1e+07$	$3.9e+06$	$1.0e+00$	4.5
e-01	3	14	1	1	14	T	$6.223754e+06$	$6.1e+06$	$1.2e+05$	$-9.3e+06$	$1.6e+06$	$1.0e+00$	2.9
e-01	4	43	1	1	43	T	$1.578787e+06$	$1.3e+06$	$2.3e+05$	$-1.6e+06$	$4.6e+05$	$1.0e+00$	1.5
e-02	5	89	1	1	89	T	$7.995442e+05$	$4.3e+05$	$3.7e+05$	$-5.2e+04$	$7.1e+04$	$1.0e+00$	6.0
e-02	6	137	1	1	137	T	$7.736097e+05$	$3.7e+05$	$4.0e+05$	$-2.1e+02$	$4.1e+03$	$1.0e+00$	1.5
e-03	7	212	1	1	212	F	$7.735055e+05$	$3.7e+05$	$4.0e+05$	$-8.1e-02$	$6.8e+01$	$1.0e+00$	1.9
	8	0	1	1	0	F	$7.735055e+05$	$3.7e+05$	$4.0e+05$	----	$1.4e-01$	$1.0e+00$	

converged : True

reason : Norm of the gradient less than tolerance

cumulative CG iterations : 503

cumulative cost evaluations : 9

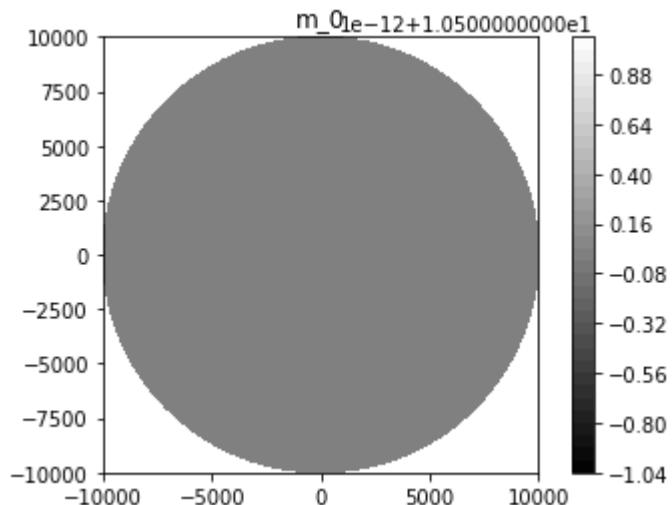
cumulative gradient evaluations : 9

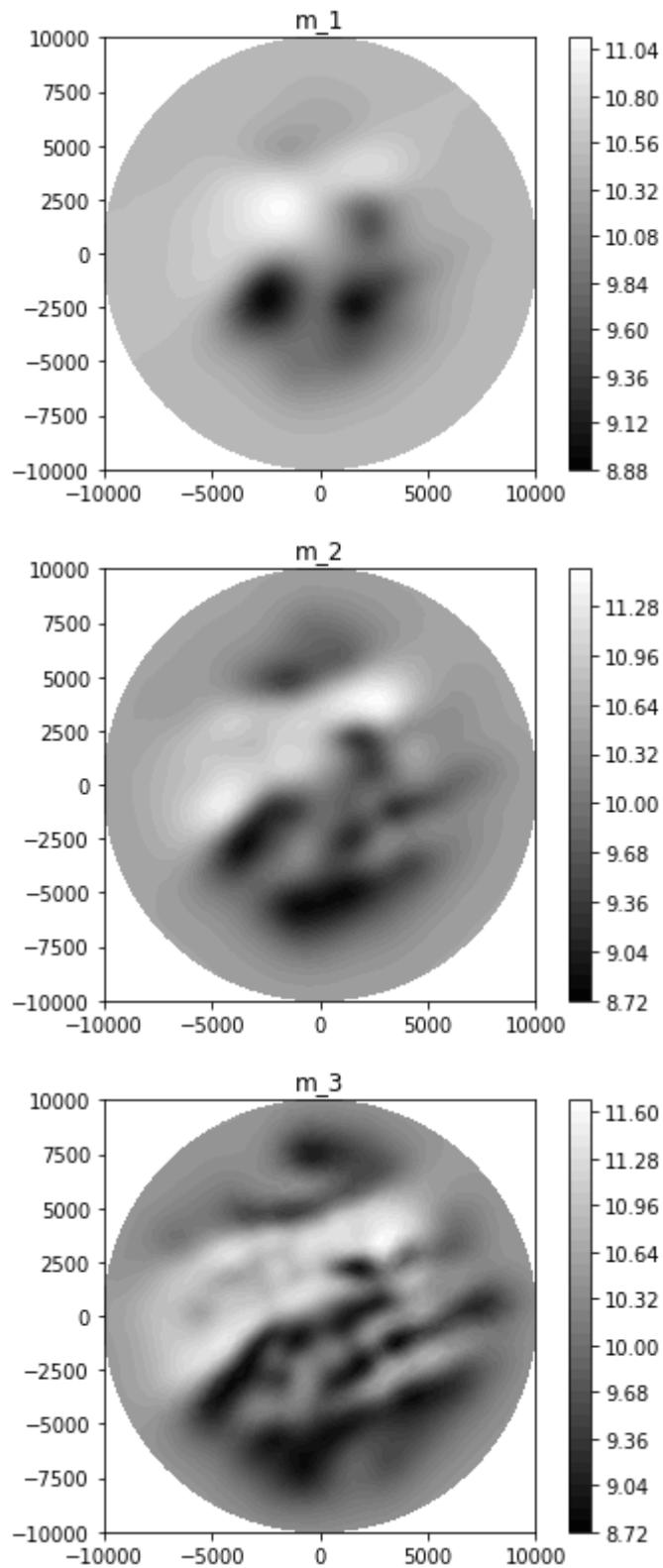
cumulative Hessian vector products (excluding preconditioner builds) : 503

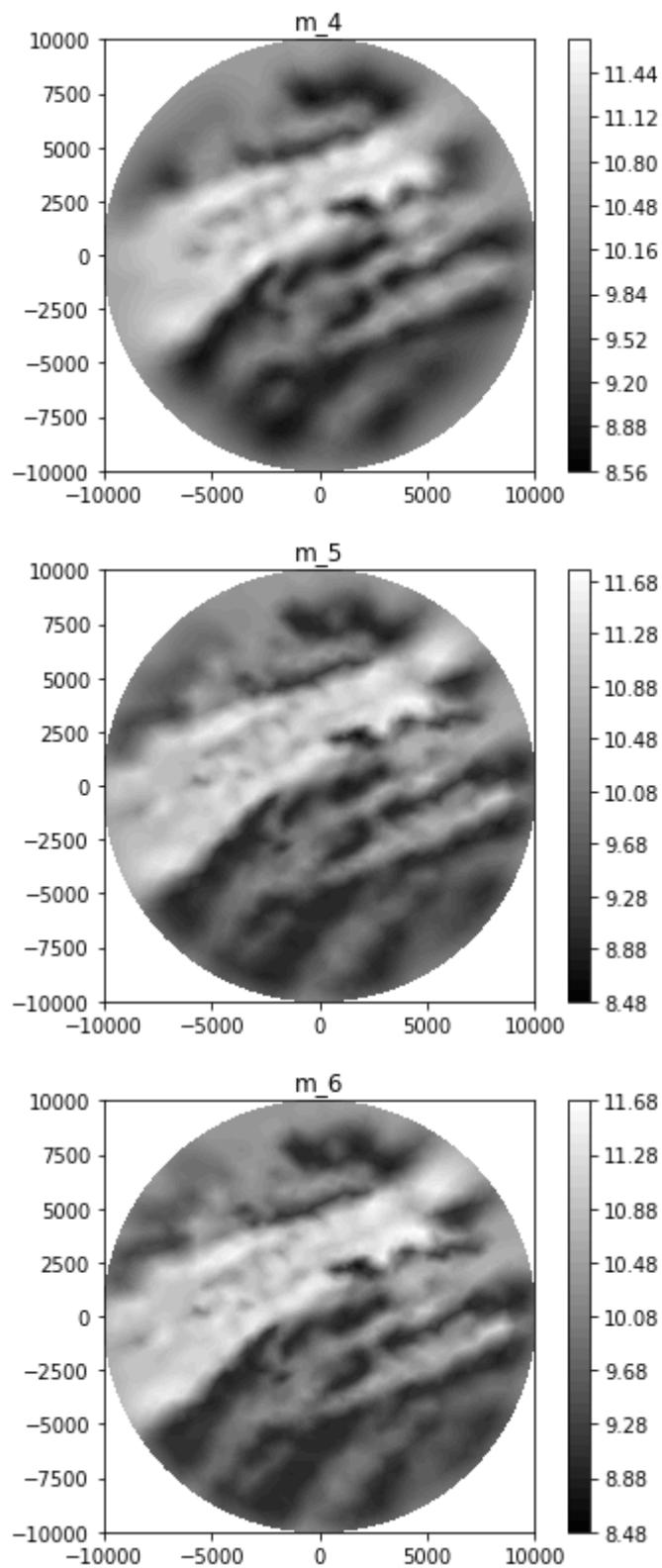
===== End Newton CG convergence information =====

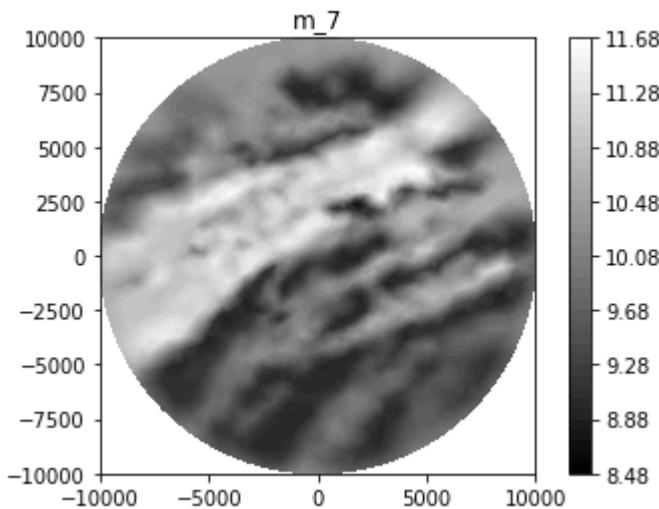
=====

gamma= 6975.784285975133 , noise_datanorm= 863.5355316705534 , misfit_datanorm= 863.5355639023904









In [35]:

```
misfit_datanorm = morozov_helper()
```

```
gamma= 6975.784285975133 , noise_datanorm= 863.5355316705534 , misfit_datanorm=
863.5355639023904
```

Plots to make (4):

- Newton CG iterations and gradient table (psf and reg)
- Newton iterates (reg and psf)
- u recovered (with arrows)

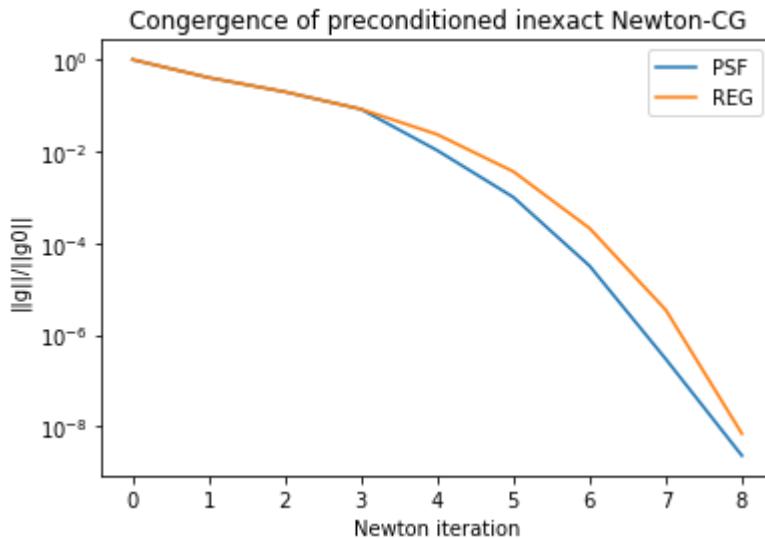
In [36]:

```
norm_grad_newton_pch = np.array(PCH_convergence_info.gradnorm)
norm_grad_newton_reg = np.array(REG_convergence_info.gradnorm)

plt.figure()
plt.semilogy(norm_grad_newton_pch/norm_grad_newton_pch[0])
plt.semilogy(norm_grad_newton_reg/norm_grad_newton_reg[0])
plt.legend(['PSF', 'REG'])
plt.xlabel('Newton iteration')
plt.ylabel('||g||/||g0||')
plt.title('Congergence of preconditioned inexact Newton-CG')
```

Out[36]:

```
Text(0.5, 1.0, 'Congergence of preconditioned inexact Newton-CG')
```



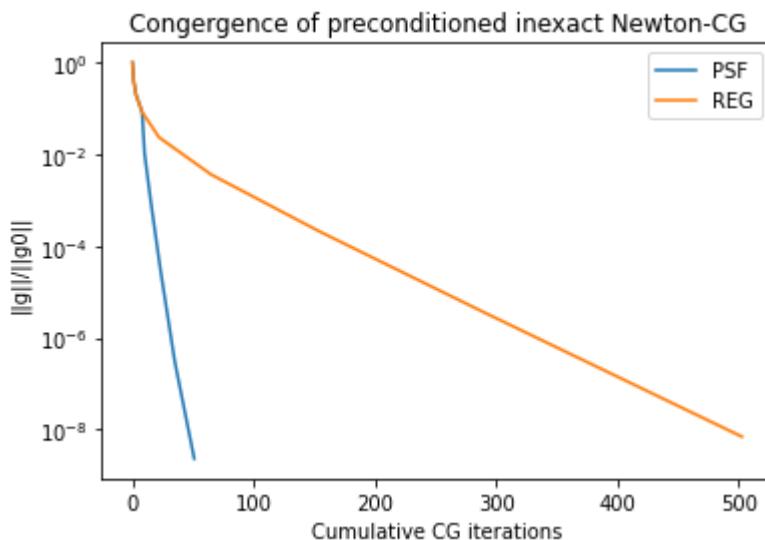
```
In [37]: cumulative_cg_pch = np.concatenate([np.array([0.0]), np.cumsum(PCH_convergence_info.hess_matvecs)])
cumulative_cg_reg = np.concatenate([np.array([0.0]), np.cumsum(REG_convergence_info.hess_matvecs)])

print('cumulative_cg_pch=', cumulative_cg_pch)
print('cumulative_cg_reg=', cumulative_cg_reg)

plt.figure()
plt.semilogy(cumulative_cg_pch, norm_grad_newton_pch/norm_grad_newton_pch[0])
plt.semilogy(cumulative_cg_reg, norm_grad_newton_reg/norm_grad_newton_reg[0])
plt.legend(['PSF', 'REG'])
plt.xlabel('Cumulative CG iterations')
plt.ylabel('||g||/||g0||')
plt.title('Convergence of preconditioned inexact Newton-CG')

cumulative_cg_pch= [ 0.  1.  3.  8. 10. 15. 23. 35. 51.]
cumulative_cg_reg= [ 0.  1.  3.  8. 22. 65. 154. 291. 503.]
```

Out[37]:



```
In [38]: print(PCH_convergence_info.hess_matvecs)
print(PCH_convergence_info.cg_iter)
```

[1, 2, 5, 2, 5, 8, 12, 16, 0]

```
[1, 2, 5, 2, 5, 8, 12, 16, 0]
```

In [39]:

```
spatialdim = 2
num_moment_matvecs = 1 + spatialdim + spatialdim * (spatialdim + 1) / 2.0
num_preconditioner_build_matvecs = num_moment_matvecs + num_batches

print('num_moment_matvecs=', num_moment_matvecs)
print('num_preconditioner_build_matvecs=', num_preconditioner_build_matvecs)

num_moment_matvecs= 6.0
num_preconditioner_build_matvecs= 11.0
```

In [40]:

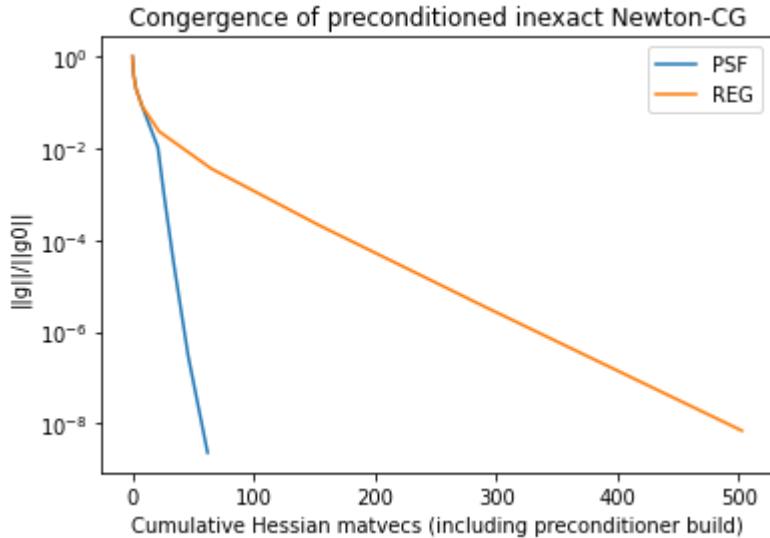
```
cumulative_hessian_matvecs_psf = [0.0]
for nh, build in zip(PCH_convergence_info.hess_matvecs[:-1], PCH_convergence_info.newton_iter):
    prev_matvecs = cumulative_hessian_matvecs_psf[-1]
    if build:
        new_matvecs = nh + num_preconditioner_build_matvecs
    else:
        new_matvecs = nh
    cumulative_hessian_matvecs_psf.append(prev_matvecs + new_matvecs)

cumulative_hessian_matvecs_psf = np.array(cumulative_hessian_matvecs_psf)
cumulative_hessian_matvecs_reg = cumulative_cg_reg
print('cumulative_hessian_matvecs_psf=', cumulative_hessian_matvecs_psf)
print('cumulative_hessian_matvecs_reg=', cumulative_hessian_matvecs_reg)

plt.figure()
plt.semilogy(cumulative_hessian_matvecs_psf, norm_grad_newton_pch/norm_grad_newton_cg)
plt.semilogy(cumulative_hessian_matvecs_reg, norm_grad_newton_reg/norm_grad_newton_cg)
plt.legend(['PSF', 'REG'])
plt.xlabel('Cumulative Hessian matvecs (including preconditioner build)')
plt.ylabel('||g||/||g0||')
plt.title('Convergence of preconditioned inexact Newton-CG')
```

```
cumulative_hessian_matvecs_psf= [ 0.  1.  3.  8. 21. 26. 34. 46. 62.]
cumulative_hessian_matvecs_reg= [ 0.  1.  3.  8. 22. 65. 154. 291. 503.]
```

Out[40]:



In [41]:

```
cumulative_stokes_psf = [PCH_convergence_info.cost_calls[0] + PCH_convergence_info.newton_iter]
for k in range(1, len(PCH_convergence_info.newton_iter)):
```

```

cs = cumulative_stokes_psf[-1]
cs += PCH_convergence_info.cost_calls[k]
cs += PCH_convergence_info.grad_calls[k]
cs += 2*PCH_convergence_info.hess_matvecs[k-1] # incremental forward and inc
if PCH_convergence_info.build_precond[k-1]:
    cs += 2*num_preconditioner_build_matvecs
cumulative_stokes_psf.append(cs)

cumulative_stokes_psf = np.array(cumulative_stokes_psf, dtype=int)
print('cumulative_stokes_psf=', cumulative_stokes_psf)

cumulative_stokes_reg = [REG_convergence_info.cost_calls[0] + REG_convergence_ir
for k in range(1,len(REG_convergence_info.newton_iter))]:
    cs = cumulative_stokes_reg[-1]
    cs += REG_convergence_info.cost_calls[k]
    cs += REG_convergence_info.grad_calls[k]
    cs += 2*REG_convergence_info.hess_matvecs[k-1] # incremental forward and inc
    cumulative_stokes_reg.append(cs)

cumulative_stokes_reg = np.array(cumulative_stokes_reg)
print('cumulative_stokes_reg=', cumulative_stokes_reg)

plt.figure()
plt.semilogy(cumulative_stokes_psf, norm_grad_newton_pch/norm_grad_newton_pch[0]
plt.semilogy(cumulative_stokes_reg, norm_grad_newton_reg/norm_grad_newton_reg[0]
plt.legend(['PSF', 'REG'])
plt.xlabel('Cumulative Stokes solves')
plt.ylabel('||g||/||g0||')
plt.title('Congergence of preconditioned inexact Newton-CG')

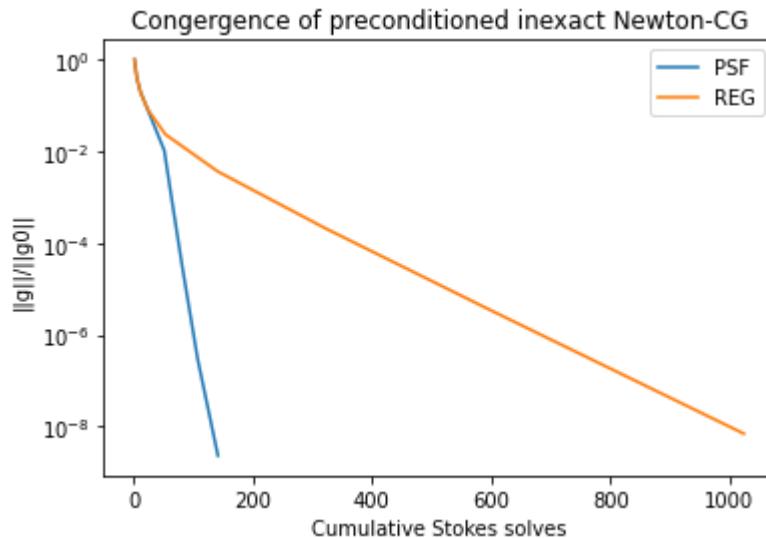
```

```

cumulative_stokes_psf= [ 2   6   12  24  52  64  82 108 142]
cumulative_stokes_reg= [   2    6   12  24   54  142  322  598 1024]
Text(0.5, 1.0, 'Congergence of preconditioned inexact Newton-CG')

```

Out[41]:



In [43]:

```

dl.File(save_dir_str + "/recovered_pressure" + str(outflow_constant)+".pvf") <-
dl.File(save_dir_str + "/recovered_velocity" + str(outflow_constant)+".pvf") <-

num_cg_newton_reg = np.array(REG_convergence_info.hess_matvecs)
num_stokes_newton_reg = (np.array(REG_convergence_info.cost_calls) +
                        np.array(REG_convergence_info.grad_calls) +
                        2*np.array(REG_convergence_info.hess_matvecs))

```

```

num_cg_newton_psf = np.array(PCH_convergence_info.hess_matvecs)
num_stokes_newton_psf = (np.array(PCH_convergence_info.cost_calls) +
                         np.array(PCH_convergence_info.grad_calls) +
                         2*np.array(PCH_convergence_info.hess_matvecs))

num_stokes_build_preconditioner = 2*num_preconditioner_build_matvecs*np.array(PC
num_stokes_build_preconditioner = np.array(num_stokes_build_preconditioner, dtype='

np.savetxt(save_dir_str + '/norm_grad_newton_reg.txt', norm_grad_newton_reg)
np.savetxt(save_dir_str + '/norm_grad_newton_psf.txt', norm_grad_newton_pch)
np.savetxt(save_dir_str + '/num_cg_newton_reg.txt', num_cg_newton_reg)
np.savetxt(save_dir_str + '/num_cg_newton_psf.txt', num_cg_newton_psf)
np.savetxt(save_dir_str + '/num_stokes_newton_reg.txt', num_stokes_newton_reg)
np.savetxt(save_dir_str + '/num_stokes_newton_psf.txt', num_stokes_newton_psf)
np.savetxt(save_dir_str + '/num_stokes_build_preconditioner.txt', num_stokes_bui

ps = " "*5 + "|" + "*10 + "PSF" + " "*10 + "|" + " "*10 + "REG" + " "*10 + "\r
ps += "-"*5 + "|" + "-"*23 + "|" + "-"*23 + "\n"
ps += "{0:>4} | {1:<3} {2:<7} {3:>9} | {4:<3} {5:<7} {6:>9}\n".format(
    "Iter", "#CG", "#Stokes", "||g||", "#CG", "#Stokes", "||g||")
for k in range(len(norm_grad_newton_reg)):
    g_psf = norm_grad_newton_pch[k]
    g_reg = norm_grad_newton_reg[k]
    ncg_psf = num_cg_newton_psf[k]
    ncg_reg = num_cg_newton_reg[k]
    ns_psf = num_stokes_newton_psf[k]
    ns_reg = num_stokes_newton_reg[k]
    nbuild = num_stokes_build_preconditioner[k]

    ps += "{:>4}".format(k)
    ps += " | "
    ps += "{:<3d}".format(ncg_psf)
    ps += " "
    if nbuild > 0:
        ps += "{:<7}".format(str(ns_psf) + " + " + str(int(nbuild)))
    else:
        ps += "{:<7d}".format(ns_psf)
    ps += " "
    ps += "{:>9}".format(np.format_float_scientific(g_psf, precision=2, trim='k'
    ps += " | "
    ps += "{:<3d}".format(ncg_reg)
    ps += " "
    ps += "{:<7d}".format(ns_reg)
    ps += " "
    ps += "{:>9}".format(np.format_float_scientific(g_reg, precision=2, trim='k'
    ps += "\n"

print(ps)

num_cg_total_psf = np.sum(num_cg_newton_psf)
print('num_cg_total_psf=', num_cg_total_psf)

num_cg_total_reg = np.sum(num_cg_newton_reg)
print('num_cg_total_reg=', num_cg_total_reg)

num_stokes_total_psf = np.sum(num_stokes_newton_psf)
print('num_stokes_total_psf=', num_stokes_total_psf)

num_stokes_total_reg = np.sum(num_stokes_newton_reg)
print('num_stokes_total_reg=', num_stokes_total_reg)

```

# Iter	#CG (PSF)	#Stokes (PSF)	g (PSF)	#CG (REG)	#Stokes (REG)
# 1	1	1234	1.95e+07	1	2345
# 2	2	1234	7.88e+06	2	2345
# 3	5	1234	3.86e+06	5	2345
# 4	2	1234 + 24	1.62e+06	14	2345
#
# 8	15	1234	3.74e+01	210	2345
# 9	---	2	1.26e-01	---	2

	PSF			REG		
Iter	#CG	#Stokes	g	#CG	#Stokes	g
0	1	4	1.95e+07	1	4	1.95e+07
1	2	6	7.91e+06	2	6	7.91e+06
2	5	12	3.87e+06	5	12	3.87e+06
3	2	6 + 22	1.62e+06	14	30	1.62e+06
4	5	12	2.05e+05	43	88	4.57e+05
5	8	18	1.96e+04	89	180	7.08e+04
6	12	26	6.24e+02	137	276	4.12e+03
7	16	34	5.82e+00	212	426	6.84e+01
8	0	2	4.54e-02	0	2	1.37e-01

```
num_cg_total_psf= 51
num_cg_total_reg= 503
num_stokes_total_psf= 120
num_stokes_total_reg= 1024
```

In [44]:

```
for k in range(len(m_iter_PCH)):
    mk_pch = dl.Function(Vh2)
    mk_pch.vector()[:] = m_iter_PCH[k]
    dl.File(save_dir_str + "/m" + str(k) + "_psf.pvd") << mk_pch

for k in range(len(m_iter_REG)):
    mk_reg = dl.Function(Vh2)
    mk_reg.vector()[:] = m_iter_REG[k]
    dl.File(save_dir_str + "/m" + str(k) + "_reg.pvd") << mk_reg
```

CG solve comparing PCH to REG to None preconditioning

In [45]:

```
all_PCK = list()
all_Hd_pch = list()
for nb in all_num_batches:
    print('building PCH' + str(nb) + ' preconditioner')
    PCKx = ProductConvolutionKernel(Vh2, Vh2,
                                      apply_misfit_gauss_newton_hessian_petsc,
                                      apply_misfit_gauss_newton_hessian_petsc,
                                      nb, nb,
                                      tau_rows=tau, tau_cols=tau,
                                      num_neighbors_rows=num_neighbors,
                                      num_neighbors_cols=num_neighbors)
    Hd_pch_nonsymx, extrasx = make_hmatrix_from_kernel(PCKx, hmatrix_tol=hmatrix_tol)

    # ----- build spd approximation of Hd
    Hd_pchx = Hd_pch_nonsymx.spd()
```

```

    all_PCK.append(PCKx)
    all_Hd_pch.append(Hd_pchx)

building PCH1 preconditioner
Computing impulse response moments
getting spatially varying volume
getting spatially varying mean
getting spatially varying covariance
Preparing sample point batch stuff
Preparing c++ object
Building initial sample point batches

Making row and column cluster treesUsing ProductConvolutionKernelRBFColsOnly!

Making block cluster trees
Building A kernel hmatrix
— building H-matrix ( tol = Making input and output mass matrix hmatrices
Computing A_hmatrix = M_out_hmatrix * A_kernel_hmatrix * M_in_hmatrix
1e-06 )
making hmatrix spd ] 7% ETA 13 s (5.30 GB)
symmetrizing
getting smallest eigenvalue with Lanczos
done in 13.55s
size of H-matrix = 4.42 MB
lambda_min= -21939.888935957897
scaling_at_lambda_min= 0.9878048780487805
scaling_at_zero= 0.5
Setting up operator T = (2*A - (b+a) I) / (b-a)
computing T^(2^k)
computing T^(2^1) = T^(2^0) * T^(2^0)

computing T^(2^2) = T^(2^1) * T^(2^1)
computing negative spectral projector Pi_minus = I / (I + T^(2^k))
— H-matrix multiplication C=A*B
[=====
done in 24.047887325286865 5.30 GB)
done in 32.00s
size of C = 5.37 MB

— H-matrix multiplication C=A*B
done in 33.39s
size of C = 8.05 MB
bytes

computing A_plus = Pi * A

— H-matrix multiplication C=A*B
[==-
building PCH5 preconditioner
Computing impul
se response moments
getting spatially varying volume
done in 38.49s
size of C = 21.30 MB
getting spatially varying mean
getting spatially varying covariance
Preparing sample point batch stuff
Preparing c++ object
Building initial sample point batches

```

Making row and column cluster treesUsing ProductConvolutionKernelRBFColsOnly!

```

Making block cluster trees
Building A kernel hmatrix
Making input and output mass matrix hmatrices
— building H-matrix ( tol = 1e-06 )
[==-
Computing A_hmatrix = M_out_hmatrix * A_kernel_hmatrix * M_in_hmatrix
] 14% ETA 12 s (5.35 GB)
symmetrizing
[=====-
    done in 14.31s
    size of H-matrix = 4.42 MB
lambda_min= -31571.25915903734
scaling_at_lambda_min= 0.9878048780487805
scaling_at_zero= 0.5
Setting up operator T = (2*A - (b+a) I) / (b-a)
computing T^(2^k)
computing T^(2^1) = T^(2^0) * T^(2^0)

— H-matrix multiplication C=A*B
] 7% ETA 22 s (5.35 GB)computing T^(2
^2) = T^(2^1) * T^(2^1)
done in 32.73s
] 15% ETA 22 s (5.35 GB)
size of C = 5.28 MB

— H-matrix multiplication C=A*B
done in 34.24s
size of C = 7.69 MB

— H-matrix inverse ( rtol = 1e-07 , atol = 1e-12 , overwrite= False )
done in 24.184748649597168
size of inverse = 17725073 bytes

computing A_plus = Pi * A

— H-matrix multiplication C=A*B
done in 39.20s
size of C = 20.57 MB
building PCH25 preconditioner
Computing impulse response moments
getting spatially varying volume
getting spatially varying mean
getting spatially varying covariance
Preparing sample point batch stuff
Preparing c++ object
Building initial sample point batches

Making row and column cluster treesUsing ProductConvolutionKernelRBFColsOnly!

Making block cluster trees
Building A kernel hmatrix
Making input and output mass matrix hmatrices
— building H-matrix ( tol = 1e-06 )
[==-
Computing A_hmatrix = M_out_hmatrix * A_kernel_hmatrix * M_in_hmatrix
making hmatrix spd
[=====-
    done in 14.98s
    size of H-matrix = 4.42 MB
lambda_min= -32200.061569526715

```

```

scaling_at_lambda_min= 0.9878048780487805
scaling_at_zero= 0.5
Setting up operator T = (2*A - (b+a) I) / (b-a)
computing T^(2^k)
computing T^(2^1) = T^(2^0) * T^(2^0)
computing T^(2^2) = T^(2^1) * T^(2^1)

— H-matrix multiplication C=A*B
done in 32.94s
size of C = 5.09 MB

— H-matrix multiplication C=A*B
[=====] computing negative spectral projector Pi_mi
nus = I / (I + T^(2^k))
done in 34.64s
size of C = 7.47 MB
— H-matrix inverse ( rtol = 1e-07 , atol = 1e-12 , overwrite= False )
[ ] 0% (5.38 GB) done in 26.092469
215393066
size of inverse = 17345569 bytes
computing absolute value projector Pi = I - 2*Pi_minus
computing A_plus = Pi * A

— H-matrix multiplication C=A*B
done in 39.24s
size of C = 20.28 MB

```

In [46]: REG.update_gamma(gamma_morozov)

```

# Rebuild reg hmatrix with same block cluster tree as PCH data misfit hmatrix
print('Building Regularization H-Matrix')
R_hmatrix = REG.make_R_hmatrix(all_Hd_pch[0].bct, rtol=1e-6)

all_H_pch = [Hd + R_hmatrix for Hd in all_Hd_pch]
all_preconditioners = [H.factorized_inverse() for H in all_H_pch]

```

```

/home/nick/anaconda3/envs/fenics4/lib/python3.9/site-packages/scipy/sparse/linalg/dsolve/linsolve.py:318: SparseEfficiencyWarning: splu requires CSC matrix format
warn('splu requires CSC matrix format', SparseEfficiencyWarning)
Building Regularization H-Matrix
matrix has dimension 1416 x 1416
no of non-zeroes      = 9746
matrix is              real valued
format                = non symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 1.1247e+06
sparsity constant = 20
size of H-matrix     = 4.06 MB
|A|_F                 = 1.1247e+06
|S-A|_2 = 3.13924e-09
matrix has dimension — H-matrix inverse ( rtol = 1e-06 , atol = 1e-14 , ove
rwrite= False )
1416 x 1416
no of non-zeroes      = 9746
matrix is              real valued
format                = symmetric
size of sparse matrix = 163.52 kB
|S|_F                 = 5.18561e+06

```

```

    sparsity constant = 20
    size of H-matrix  = 2.29 MB
    |A|_F             = 5.18561e+06
|S-A|_2 = 4.66653e-09
done in 15.561947107315063
size of inverse = 3361874 bytes

— H-matrix multiplication C=A*B
done in 25.59s
size of C = 3.88 MB

— H-matrix multiplication C=A*B
done in 35.28s
size of C = 8.54 MB

— LU factorisation ( rtol = 1e-07 )
done in 34.12s
size of LU factor = 23.06 MB

— LU factorisation ( rtol = 1e-07 )
done in 33.98s
size of LU factor = 22.16 MB

— LU factorisation ( rtol = 1e-07 )
done in 34.10s
size of LU factor = 21.89 MB

```

B)

In [50]:

```

def cg_wrapper(A_linop, b, M_linop, tol=1e-10, x_true=None):
    if x_true is None:
        x_true = np.nan * np.zeros(b.shape)
    all_relres = list()
    all_err = list()

    ii_list = [0]
    def callback(xk):
        relres = np.linalg.norm(b - A_linop.matvec(xk)) / np.linalg.norm(b)
        all_relres.append(relres)
        err = np.linalg.norm(xk - x_true) / np.linalg.norm(x_true)
        all_err.append(err)
        print('ii=', ii_list[0], ', relres=', relres, ', err=', err)
        ii_list[0] += 1

    x, info = spla.cg(A_linop, b, M=M_linop, tol=tol, callback=callback, maxiter=100)
    all_relres = np.array(all_relres)
    all_err = np.array(all_err)
    return x, info, all_relres, all_err

```

In [48]:

```

H_linop = spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=apply_hessian)
Hgn_linop = spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=apply_gauss_newton)
ireg_linop = spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=REG.solve_hessian)
all_p_linop = [spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=p.matvec) for p in p_list]

b = np.random.randn(Vh2.dim())

print('GN: get truth:')
x_true_gn, _, _, _ = cg_wrapper(Hgn_linop, b, all_p_linop[-1])

all_relres_gn = list()

```

```

all_errs_gn = list()
for nb, p_linop in zip(all_num_batches, all_p_linop):
    print('GN: psf'+str(nb))
    _, _, relres_gn, errs_gn = cg_wrapper(Hgn_linop, b, p_linop, x_true=x_true_c)
    all_relres_gn.append(relres_gn)
    all_errs_gn.append(errs_gn)

print('GN: reg')
_, _, relres_gn_reg, errs_gn_reg = cg_wrapper(Hgn_linop, b, ireg_linop, x_true=x_true_c)

print('GN: none')
_, _, relres_gn_none, errs_gn_none = cg_wrapper(Hgn_linop, b, None, x_true=x_true_c)

plt.semilogy(errs_gn_none[:-1])
plt.semilogy(errs_gn_reg[:-1])
for errs_gn in all_errs_gn:
    plt.semilogy(errs_gn[:-1])
plt.title('Error vs. Krylov iteration: Gauss-Newton')
plt.xlabel('Iteration k')
plt.ylabel('relative error')
plt.legend(['None', 'REG'] + ['PSF'+str(nb) for nb in all_num_batches])

plt.figure()
plt.semilogy(relres_gn_none)
plt.semilogy(relres_gn_reg)
for relres_gn in all_relres_gn:
    plt.semilogy(relres_gn)
plt.title('Residual vs. Krylov iteration: Gauss-Newton')
plt.xlabel('Iteration k')
plt.ylabel('relative residual')
plt.legend(['None', 'REG'] + ['PSF'+str(nb) for nb in all_num_batches])

```

GN: get truth:

```

ii= 0 , relres= 0.3176683818809919 , err= nan
ii= 1 , relres= 0.12219419583314826 , err= nan
ii= 2 , relres= 0.05929028993565218 , err= nan
ii= 3 , relres= 0.029223360826238765 , err= nan
ii= 4 , relres= 0.014927410112437286 , err= nan
ii= 5 , relres= 0.008249646977940985 , err= nan
ii= 6 , relres= 0.004492700717803115 , err= nan
ii= 7 , relres= 0.002434905698076133 , err= nan
ii= 8 , relres= 0.0013472210970393488 , err= nan
ii= 9 , relres= 0.0006446787123482165 , err= nan
ii= 10 , relres= 0.000331846225939158 , err= nan
ii= 11 , relres= 0.00015445681454121193 , err= nan
ii= 12 , relres= 7.087694748843961e-05 , err= nan
ii= 13 , relres= 3.473889744332784e-05 , err= nan
ii= 14 , relres= 1.4946417517103958e-05 , err= nan
ii= 15 , relres= 7.4650943849711214e-06 , err= nan
ii= 16 , relres= 3.6027466418852616e-06 , err= nan
ii= 17 , relres= 1.6374987115869782e-06 , err= nan
ii= 18 , relres= 7.260859931916766e-07 , err= nan
ii= 19 , relres= 3.1071291058666855e-07 , err= nan
ii= 20 , relres= 1.362927040644032e-07 , err= nan
ii= 21 , relres= 5.8710800599917436e-08 , err= nan
ii= 22 , relres= 2.4995763683535774e-08 , err= nan
ii= 23 , relres= 1.171649291665167e-08 , err= nan
ii= 24 , relres= 4.8960399468540766e-09 , err= nan
ii= 25 , relres= 1.8426316010750442e-09 , err= nan
ii= 26 , relres= 6.992902245448452e-10 , err= nan
ii= 27 , relres= 2.8468496756831994e-10 , err= nan

```

```

ii= 28 , relres= 1.2942732288445515e-10 , err= nan
ii= 29 , relres= 6.771541589629453e-11 , err= nan
GN: psf1
ii= 0 , relres= 0.4303236626802175 , err= 0.2717020210717325
ii= 1 , relres= 0.2555265124959215 , err= 0.12252313046023418
ii= 2 , relres= 0.14672933088858342 , err= 0.06775500624745748
ii= 3 , relres= 0.09387532650646126 , err= 0.03694507657155087
ii= 4 , relres= 0.05749066166412285 , err= 0.024681849207210088
ii= 5 , relres= 0.03536503325229412 , err= 0.01414281762348893
ii= 6 , relres= 0.023291299659775777 , err= 0.008928124218214066
ii= 7 , relres= 0.014891503622148458 , err= 0.005704552293839755
ii= 8 , relres= 0.009680569051942728 , err= 0.003412379371716328
ii= 9 , relres= 0.00620422379816278 , err= 0.00216622079075269
ii= 10 , relres= 0.004020462044569462 , err= 0.0012934516287395888
ii= 11 , relres= 0.002424874462499395 , err= 0.0007960616330352545
ii= 12 , relres= 0.0014979762685737301 , err= 0.0004660584158228963
ii= 13 , relres= 0.0008939464212390416 , err= 0.00029481167226135386
ii= 14 , relres= 0.0005391273464371393 , err= 0.0001858644088597927
ii= 15 , relres= 0.0003422553740079775 , err= 0.00011192514957659832
ii= 16 , relres= 0.00021207179816944643 , err= 7.699028415455345e-05
ii= 17 , relres= 0.00013239873720282807 , err= 4.773709776935418e-05
ii= 18 , relres= 8.228939438767952e-05 , err= 2.8363142281393274e-05
ii= 19 , relres= 4.769484675429377e-05 , err= 1.8083419494714644e-05
ii= 20 , relres= 2.9204278233260142e-05 , err= 9.04493560396812e-06
ii= 21 , relres= 1.7110497475221808e-05 , err= 6.115674690108275e-06
ii= 22 , relres= 9.650734568833551e-06 , err= 3.4067583315337225e-06
ii= 23 , relres= 5.8191751717731e-06 , err= 2.1940663286548443e-06
ii= 24 , relres= 3.5687023738934375e-06 , err= 1.3673827622071813e-06
ii= 25 , relres= 2.2087218797760863e-06 , err= 8.100138806098674e-07
ii= 26 , relres= 1.422394049685037e-06 , err= 4.739309811416228e-07
ii= 27 , relres= 8.743639694047915e-07 , err= 3.026299399956638e-07
ii= 28 , relres= 5.214805395438914e-07 , err= 1.6665609245351856e-07
ii= 29 , relres= 3.012419089327925e-07 , err= 1.0535178492941126e-07
ii= 30 , relres= 1.6027321874843957e-07 , err= 5.932240133183649e-08
ii= 31 , relres= 9.474400335824318e-08 , err= 2.979386630636426e-08
ii= 32 , relres= 5.593200567609568e-08 , err= 1.9232650037924314e-08
ii= 33 , relres= 3.091973074006108e-08 , err= 1.0578363647943844e-08
ii= 34 , relres= 1.7951443503208012e-08 , err= 5.890671617420606e-09
ii= 35 , relres= 1.0096109499473212e-08 , err= 3.640301706989164e-09
ii= 36 , relres= 6.5552737332496135e-09 , err= 2.1331008329346033e-09
ii= 37 , relres= 4.322184530407875e-09 , err= 1.4591763262835361e-09
ii= 38 , relres= 2.6846461348090248e-09 , err= 8.968458265824674e-10
ii= 39 , relres= 1.728313525691744e-09 , err= 5.355869909418848e-10
ii= 40 , relres= 9.221454397331823e-10 , err= 3.141224497095239e-10
ii= 41 , relres= 4.756205669793359e-10 , err= 1.615457063718629e-10
ii= 42 , relres= 2.70531236399651e-10 , err= 9.686271284055208e-11
ii= 43 , relres= 1.5650947281481934e-10 , err= 5.4713383236149084e-11
ii= 44 , relres= 1.041729472145198e-10 , err= 3.2004823493610865e-11
ii= 45 , relres= 6.70098045127641e-11 , err= 2.5165472434190912e-11
GN: psf5
ii= 0 , relres= 0.4011092431371095 , err= 0.2893812407776159
ii= 1 , relres= 0.1890284813593398 , err= 0.08535446932302092
ii= 2 , relres= 0.12482077844273369 , err= 0.04542358676473394
ii= 3 , relres= 0.06941615825608709 , err= 0.02990196295505023
ii= 4 , relres= 0.03932048727482909 , err= 0.015404818615934344
ii= 5 , relres= 0.027527567593108457 , err= 0.008502118209189155
ii= 6 , relres= 0.0160073765645773 , err= 0.005653201176890141
ii= 7 , relres= 0.009806627424953349 , err= 0.003599797836124551
ii= 8 , relres= 0.006037427746509738 , err= 0.0021424523963990126
ii= 9 , relres= 0.0036193142095174236 , err= 0.0013113638392699824
ii= 10 , relres= 0.002023761045460334 , err= 0.0006979901951075163

```

```

ii= 11 , relres= 0.001198432211879224 , err= 0.00036648046258855153
ii= 12 , relres= 0.0006798127403473902 , err= 0.0002361519328738246
ii= 13 , relres= 0.0003640274038694264 , err= 0.00013082968349142782
ii= 14 , relres= 0.00020271523448636575 , err= 7.364012682890258e-05
ii= 15 , relres= 0.00011239780290271259 , err= 4.1419863855806154e-05
ii= 16 , relres= 6.280887143496206e-05 , err= 2.2459431228097592e-05
ii= 17 , relres= 3.68870363205681e-05 , err= 1.2263878753017295e-05
ii= 18 , relres= 2.0995643739567486e-05 , err= 7.337221822804505e-06
ii= 19 , relres= 1.2642029051025374e-05 , err= 4.217530843548029e-06
ii= 20 , relres= 6.854541814701528e-06 , err= 2.410726536395615e-06
ii= 21 , relres= 3.698805873227959e-06 , err= 1.224803109954489e-06
ii= 22 , relres= 1.818896904636552e-06 , err= 6.408609687011261e-07
ii= 23 , relres= 9.665566419339014e-07 , err= 3.186821215252854e-07
ii= 24 , relres= 4.930141582176686e-07 , err= 1.6375237832038025e-07
ii= 25 , relres= 3.061142801516709e-07 , err= 9.010701855263016e-08
ii= 26 , relres= 1.9875157574325685e-07 , err= 5.5313044229807776e-08
ii= 27 , relres= 1.0521116151025253e-07 , err= 3.6003033108737843e-08
ii= 28 , relres= 6.249452662583066e-08 , err= 1.921702096831036e-08
ii= 29 , relres= 3.690709220219193e-08 , err= 1.1231712990827585e-08
ii= 30 , relres= 2.154831454077728e-08 , err= 6.555872804673522e-09
ii= 31 , relres= 1.0689509895331447e-08 , err= 3.6802332309686628e-09
ii= 32 , relres= 5.829928125485472e-09 , err= 1.994203821391325e-09
ii= 33 , relres= 2.967362986062361e-09 , err= 1.0111126848797098e-09
ii= 34 , relres= 1.5708865094066604e-09 , err= 5.374300370604884e-10
ii= 35 , relres= 9.079545077723813e-10 , err= 2.895145668323176e-10
ii= 36 , relres= 5.559844088915752e-10 , err= 1.6433358850787509e-10
ii= 37 , relres= 3.0652229575559607e-10 , err= 8.895060199317302e-11
ii= 38 , relres= 1.727363804489731e-10 , err= 5.350021885712507e-11
ii= 39 , relres= 1.3446967290053136e-10 , err= 3.283248335942117e-11
ii= 40 , relres= 5.55830706470899e-11 , err= 2.4867307832569578e-11
GN: psf25
ii= 0 , relres= 0.3176683818809919 , err= 0.2879437846256377
ii= 1 , relres= 0.12219419583314826 , err= 0.0759612740204225
ii= 2 , relres= 0.05929028993565218 , err= 0.03131283852578396
ii= 3 , relres= 0.029223360826238765 , err= 0.014862706032221243
ii= 4 , relres= 0.014927410112437286 , err= 0.006403001791535952
ii= 5 , relres= 0.008249646977940985 , err= 0.0031280115650174794
ii= 6 , relres= 0.004492700717803115 , err= 0.001633053730826282
ii= 7 , relres= 0.002434905698076133 , err= 0.0008439490463142432
ii= 8 , relres= 0.0013472210970393488 , err= 0.0004894184597691512
ii= 9 , relres= 0.0006446787123482165 , err= 0.00024671588380486496
ii= 10 , relres= 0.000331846225939158 , err= 0.00011879905119558247
ii= 11 , relres= 0.00015445681454121193 , err= 6.525943944131544e-05
ii= 12 , relres= 7.087694748843961e-05 , err= 2.7236768827535596e-05
ii= 13 , relres= 3.473889744332784e-05 , err= 1.2946246709859276e-05
ii= 14 , relres= 1.4946417517103958e-05 , err= 6.415070114685648e-06
ii= 15 , relres= 7.4650943849711214e-06 , err= 3.0087836904132727e-06
ii= 16 , relres= 3.6027466418852616e-06 , err= 1.4697016292211471e-06
ii= 17 , relres= 1.6374987115869782e-06 , err= 6.708798713046867e-07
ii= 18 , relres= 7.260859931916766e-07 , err= 3.118976884896549e-07
ii= 19 , relres= 3.1071291058666855e-07 , err= 1.4634035996840605e-07
ii= 20 , relres= 1.362927040644032e-07 , err= 6.404465257334846e-08
ii= 21 , relres= 5.8710800599917436e-08 , err= 2.4665296162089318e-08
ii= 22 , relres= 2.4995763683535774e-08 , err= 9.863371498652144e-09
ii= 23 , relres= 1.171649291665167e-08 , err= 4.315370883517065e-09
ii= 24 , relres= 4.8960399468540766e-09 , err= 1.8142295706086422e-09
ii= 25 , relres= 1.8426316010750442e-09 , err= 7.669658425637473e-10
ii= 26 , relres= 6.992902245448452e-10 , err= 3.0591840780331654e-10
ii= 27 , relres= 2.8468496756831994e-10 , err= 1.2118373834674516e-10
ii= 28 , relres= 1.2942732288445515e-10 , err= 5.7438182448322193e-11
ii= 29 , relres= 6.771541589629453e-11 , err= 0.0

```

```
GN: reg
ii= 0 , relres= 1.0769002777012475 , err= 0.9957936422350884
ii= 1 , relres= 1.0630500817779598 , err= 0.9908436231741391
ii= 2 , relres= 1.1141602758726303 , err= 0.9822012506930425
ii= 3 , relres= 1.1746827560368638 , err= 0.9682070866331791
ii= 4 , relres= 1.161009434508783 , err= 0.9562773651981935
ii= 5 , relres= 1.1731684056963967 , err= 0.941449515483825
ii= 6 , relres= 1.1835511098312823 , err= 0.9289060181980805
ii= 7 , relres= 1.184296588885388 , err= 0.9103822274398686
ii= 8 , relres= 1.1986593900927554 , err= 0.8954974896871912
ii= 9 , relres= 1.1382920823082465 , err= 0.8741527205692677
ii= 10 , relres= 1.1855075975890499 , err= 0.8574874758635782
ii= 11 , relres= 1.2020768086320035 , err= 0.8401137628008769
ii= 12 , relres= 1.1798891929774358 , err= 0.8124216431601815
ii= 13 , relres= 1.1845685718695276 , err= 0.7878712033554752
ii= 14 , relres= 1.1520369928108807 , err= 0.7679585767565098
ii= 15 , relres= 1.1314144847456775 , err= 0.7443275626686383
ii= 16 , relres= 1.1444017548727583 , err= 0.7243471867442297
ii= 17 , relres= 1.1161674842383869 , err= 0.7028220744708269
ii= 18 , relres= 1.0617426221772315 , err= 0.6742265969614678
ii= 19 , relres= 1.1095617333466878 , err= 0.6468820064501111
ii= 20 , relres= 1.058513867449689 , err= 0.6234870130925037
ii= 21 , relres= 1.0740759370918127 , err= 0.6022858990618729
ii= 22 , relres= 0.9904199717516234 , err= 0.5815150780381191
ii= 23 , relres= 0.9952936213434932 , err= 0.5670977247271689
ii= 24 , relres= 0.9836466269565384 , err= 0.5571468282784592
ii= 25 , relres= 0.9686058150631685 , err= 0.5427888644065186
ii= 26 , relres= 0.9758788314661947 , err= 0.5215940485958869
ii= 27 , relres= 0.9334406171204925 , err= 0.4969103180800476
ii= 28 , relres= 0.920087395078146 , err= 0.4807561456415052
ii= 29 , relres= 0.883493274953959 , err= 0.471450203145917
ii= 30 , relres= 0.8216657247278043 , err= 0.4602229580670997
ii= 31 , relres= 0.8158142915430122 , err= 0.4466817949189677
ii= 32 , relres= 0.8361572053532779 , err= 0.432783843000396
ii= 33 , relres= 0.7888722823796868 , err= 0.42437847045108795
ii= 34 , relres= 0.7693705110758545 , err= 0.4084889205109059
ii= 35 , relres= 0.7337351443987759 , err= 0.39669345571304976
ii= 36 , relres= 0.7268467633372749 , err= 0.38764073627169576
ii= 37 , relres= 0.6863023592726113 , err= 0.3799654247186039
ii= 38 , relres= 0.6990415842247668 , err= 0.36503952242555987
ii= 39 , relres= 0.6646117461306269 , err= 0.3506457351083975
ii= 40 , relres= 0.6849393606673556 , err= 0.3361500936828117
ii= 41 , relres= 0.6833570504333685 , err= 0.3271216849968181
ii= 42 , relres= 0.6526040991765805 , err= 0.3152155449382136
ii= 43 , relres= 0.6439488490042904 , err= 0.3025006486430901
ii= 44 , relres= 0.6350300407671352 , err= 0.2975820316038346
ii= 45 , relres= 0.6100329166252891 , err= 0.28638562760656433
ii= 46 , relres= 0.594511849480589 , err= 0.2722491243344326
ii= 47 , relres= 0.5449493375362268 , err= 0.2554277158818123
ii= 48 , relres= 0.57211478863479 , err= 0.2484198314348211
ii= 49 , relres= 0.5277388739621678 , err= 0.2397929558376074
ii= 50 , relres= 0.5532047317338116 , err= 0.23302637340783902
ii= 51 , relres= 0.5040298885481981 , err= 0.2247118452117809
ii= 52 , relres= 0.47483702670536926 , err= 0.21484509767028737
ii= 53 , relres= 0.5054473218353983 , err= 0.2108418147659615
ii= 54 , relres= 0.4452323113329209 , err= 0.20272206961028244
ii= 55 , relres= 0.4248298881202767 , err= 0.19287869441875027
ii= 56 , relres= 0.4157882194000672 , err= 0.18480607696534762
ii= 57 , relres= 0.39716028255849334 , err= 0.18029616443447882
ii= 58 , relres= 0.4051530409828525 , err= 0.17453403546737062
ii= 59 , relres= 0.38533481720840046 , err= 0.1706084001196475
```

```
ii= 60 , relres= 0.36835180987230914 , err= 0.16283921277943877
ii= 61 , relres= 0.35233282325751053 , err= 0.155623612237113
ii= 62 , relres= 0.34447539331640536 , err= 0.15276997276816376
ii= 63 , relres= 0.3490833569892396 , err= 0.15039479850696552
ii= 64 , relres= 0.31740980702918403 , err= 0.1441507297304453
ii= 65 , relres= 0.3110955990607711 , err= 0.13704413896162795
ii= 66 , relres= 0.3072742582736832 , err= 0.1310905681049449
ii= 67 , relres= 0.28551582822523 , err= 0.12797115848729243
ii= 68 , relres= 0.28120550361608765 , err= 0.12231429227417472
ii= 69 , relres= 0.26451965725539284 , err= 0.12024733664132303
ii= 70 , relres= 0.2718983791297826 , err= 0.11631049296030153
ii= 71 , relres= 0.252398946987641 , err= 0.1143377778579101
ii= 72 , relres= 0.2657776168195111 , err= 0.11031881121697673
ii= 73 , relres= 0.23448731839159223 , err= 0.10608640958240419
ii= 74 , relres= 0.23597025368148145 , err= 0.10472187858355421
ii= 75 , relres= 0.22560187106142804 , err= 0.10082679440138946
ii= 76 , relres= 0.20740091697888166 , err= 0.09629711118571613
ii= 77 , relres= 0.21583194270356065 , err= 0.09362231481118466
ii= 78 , relres= 0.20317954889848416 , err= 0.08894320576957862
ii= 79 , relres= 0.19158625484704953 , err= 0.08450284051148997
ii= 80 , relres= 0.19244497362850216 , err= 0.08282107430383465
ii= 81 , relres= 0.20251493511736315 , err= 0.07976045313415643
ii= 82 , relres= 0.1881261178755438 , err= 0.07749658712756238
ii= 83 , relres= 0.17735089205612928 , err= 0.07607524929769835
ii= 84 , relres= 0.18152972010442087 , err= 0.07410743824155773
ii= 85 , relres= 0.16726594430297712 , err= 0.07189804750471174
ii= 86 , relres= 0.1630542241255837 , err= 0.0682911864975478
ii= 87 , relres= 0.15735558011913742 , err= 0.06638605715111963
ii= 88 , relres= 0.15246552515007142 , err= 0.06290895280556191
ii= 89 , relres= 0.14432318524079582 , err= 0.06021992282267252
ii= 90 , relres= 0.15049163440236937 , err= 0.058732392298771446
ii= 91 , relres= 0.14758064154070766 , err= 0.05733275981999987
ii= 92 , relres= 0.12942615905224641 , err= 0.05472298714218963
ii= 93 , relres= 0.132314149652797 , err= 0.05337985720155578
ii= 94 , relres= 0.13303698622424828 , err= 0.05217634321394923
ii= 95 , relres= 0.11821759338205788 , err= 0.0506317666893921
ii= 96 , relres= 0.11549379923935926 , err= 0.04812024379424628
ii= 97 , relres= 0.11396366554089404 , err= 0.04736708869391249
ii= 98 , relres= 0.11557555152290747 , err= 0.04569793912308334
ii= 99 , relres= 0.10296131241738765 , err= 0.04370651432598896
ii= 100 , relres= 0.09790859438123548 , err= 0.041229683243582047
ii= 101 , relres= 0.09858920916782567 , err= 0.04067662594533005
ii= 102 , relres= 0.10077568124388507 , err= 0.03944868642474173
ii= 103 , relres= 0.08822517475709692 , err= 0.03788151855788046
ii= 104 , relres= 0.09145716953512965 , err= 0.03617549782736667
ii= 105 , relres= 0.08497224786420923 , err= 0.03547227292572589
ii= 106 , relres= 0.08198026660193972 , err= 0.03461383982367663
ii= 107 , relres= 0.08115962366309304 , err= 0.03314777928385781
ii= 108 , relres= 0.07902373451249521 , err= 0.03244557738599367
ii= 109 , relres= 0.07678717015656078 , err= 0.03101630993705234
ii= 110 , relres= 0.07395788166019839 , err= 0.030056719563953137
ii= 111 , relres= 0.07971755701327624 , err= 0.029401586119991854
ii= 112 , relres= 0.069219005501756 , err= 0.028143726068118304
ii= 113 , relres= 0.07232008758605968 , err= 0.02714150702795732
ii= 114 , relres= 0.06628345175945217 , err= 0.026261115601319752
ii= 115 , relres= 0.06365458583977751 , err= 0.025956257095115856
ii= 116 , relres= 0.06325759885464834 , err= 0.02462270776821437
ii= 117 , relres= 0.059084512050663604 , err= 0.024088679139243358
ii= 118 , relres= 0.05894994112298424 , err= 0.023566727014444205
ii= 119 , relres= 0.05362355157698197 , err= 0.02219678715488902
ii= 120 , relres= 0.05515266363431787 , err= 0.021111358460305884
```

```
ii= 121 , relres= 0.053840592505646705 , err= 0.020803232507617606
ii= 122 , relres= 0.05221400856951298 , err= 0.02022055191393713
ii= 123 , relres= 0.048137595194874094 , err= 0.01889124971413155
ii= 124 , relres= 0.0485248474874924 , err= 0.018534657521570914
ii= 125 , relres= 0.04784952379881909 , err= 0.017833425123111323
ii= 126 , relres= 0.04536010198661076 , err= 0.01746001094342627
ii= 127 , relres= 0.046751104903236494 , err= 0.01660791245447781
ii= 128 , relres= 0.04265877355253833 , err= 0.016256108809464882
ii= 129 , relres= 0.041330623505660816 , err= 0.015751438491255165
ii= 130 , relres= 0.039191979179357864 , err= 0.014912046706640425
ii= 131 , relres= 0.037880132937703974 , err= 0.014762587969357312
ii= 132 , relres= 0.03766631303448826 , err= 0.014216701760181225
ii= 133 , relres= 0.033408772110990664 , err= 0.01381396088828671
ii= 134 , relres= 0.03597379215343141 , err= 0.01357854374002624
ii= 135 , relres= 0.034555391899644045 , err= 0.013210454907760526
ii= 136 , relres= 0.032759079070896896 , err= 0.012627174721900688
ii= 137 , relres= 0.03362417684598211 , err= 0.01200038410411778
ii= 138 , relres= 0.028971518852088964 , err= 0.011554726600230361
ii= 139 , relres= 0.03231989887804237 , err= 0.011119979248249383
ii= 140 , relres= 0.02887544913096215 , err= 0.010869335954484829
ii= 141 , relres= 0.027015955391680356 , err= 0.010477154513592414
ii= 142 , relres= 0.027793016694271956 , err= 0.010036285967762007
ii= 143 , relres= 0.025403897905850913 , err= 0.00972588797623392
ii= 144 , relres= 0.026449567033554295 , err= 0.009508046788962062
ii= 145 , relres= 0.024012805945478888 , err= 0.008914392115237462
ii= 146 , relres= 0.0233169556250821 , err= 0.008759979230417738
ii= 147 , relres= 0.02242522735558872 , err= 0.008189896195539939
ii= 148 , relres= 0.022127034725263686 , err= 0.008000585659573158
ii= 149 , relres= 0.021283436117198545 , err= 0.007835327172284422
ii= 150 , relres= 0.020523445006852353 , err= 0.007618094195550927
ii= 151 , relres= 0.02048607090536302 , err= 0.007415505542285955
ii= 152 , relres= 0.018740075580090096 , err= 0.007284465386201265
ii= 153 , relres= 0.019575378171047316 , err= 0.0071118334322591145
ii= 154 , relres= 0.018116492973547473 , err= 0.006759996788858534
ii= 155 , relres= 0.01723916784803538 , err= 0.006691694523998474
ii= 156 , relres= 0.018232212365225185 , err= 0.006512857137644879
ii= 157 , relres= 0.016640307700771042 , err= 0.0062761051809154065
ii= 158 , relres= 0.01594572046197494 , err= 0.0060739183376267575
ii= 159 , relres= 0.01694354010352537 , err= 0.005864365707721026
ii= 160 , relres= 0.014763719893947872 , err= 0.00558213618099711
ii= 161 , relres= 0.014451617017945496 , err= 0.0054952658201971784
ii= 162 , relres= 0.014696235121950995 , err= 0.005411166588514181
ii= 163 , relres= 0.013234247225148193 , err= 0.005116087914998213
ii= 164 , relres= 0.01290602225656306 , err= 0.004927719173085534
ii= 165 , relres= 0.01213070994694791 , err= 0.0047260715095285
ii= 166 , relres= 0.011543661229438496 , err= 0.0045826251409350005
ii= 167 , relres= 0.010795060059775305 , err= 0.004389691025681344
ii= 168 , relres= 0.011317355021860409 , err= 0.00419031562678027
ii= 169 , relres= 0.01079236546993736 , err= 0.004125242011505096
ii= 170 , relres= 0.01032512657722008 , err= 0.004010947999794697
ii= 171 , relres= 0.010301072396741277 , err= 0.003884761436561773
ii= 172 , relres= 0.01037478832899243 , err= 0.0037954972682977387
ii= 173 , relres= 0.009192913439336762 , err= 0.0036787221089439257
ii= 174 , relres= 0.009450517212919302 , err= 0.0036357146244826617
ii= 175 , relres= 0.008892914499174464 , err= 0.0034558088348988773
ii= 176 , relres= 0.008902847266040253 , err= 0.0033830090822459845
ii= 177 , relres= 0.008914081133750036 , err= 0.0032695317582100334
ii= 178 , relres= 0.009098601135714363 , err= 0.0031514970654214825
ii= 179 , relres= 0.007968678725613448 , err= 0.0030202920414251623
ii= 180 , relres= 0.008185984039901199 , err= 0.0029810652203395853
ii= 181 , relres= 0.007214638785069713 , err= 0.002820140307610544
```

```
ii= 182 , relres= 0.0074509924907323714 , err= 0.0027652936156205743
ii= 183 , relres= 0.006927940819946009 , err= 0.0026816683848745523
ii= 184 , relres= 0.006569520825729041 , err= 0.0025797417945920757
ii= 185 , relres= 0.006536557631755645 , err= 0.0025124641950664
ii= 186 , relres= 0.006666031627173326 , err= 0.0023973100688614835
ii= 187 , relres= 0.00600469252136391 , err= 0.002310443711732066
ii= 188 , relres= 0.005970373901157749 , err= 0.0021815464729703824
ii= 189 , relres= 0.005859709348136214 , err= 0.002148686155904214
ii= 190 , relres= 0.005644372302655698 , err= 0.0020848901058250057
ii= 191 , relres= 0.005350425910262472 , err= 0.0020367886244883667
ii= 192 , relres= 0.0052628588146872955 , err= 0.0019316266453294959
ii= 193 , relres= 0.00519838114345228 , err= 0.0019079613991459777
ii= 194 , relres= 0.005114358806933577 , err= 0.0018565481516815314
ii= 195 , relres= 0.004752789127254773 , err= 0.0017932835720808188
ii= 196 , relres= 0.004681970161199016 , err= 0.00176141974968891
ii= 197 , relres= 0.004871686038692451 , err= 0.001728185098710349
ii= 198 , relres= 0.004472541158631645 , err= 0.001631839205280676
ii= 199 , relres= 0.004345688014394834 , err= 0.0016031389738949322
ii= 200 , relres= 0.004479879451625316 , err= 0.0015395402130815787
ii= 201 , relres= 0.00419119181519681 , err= 0.0015188208757078968
ii= 202 , relres= 0.003828762025715958 , err= 0.0014571315920003675
ii= 203 , relres= 0.003741373762662893 , err= 0.0014192939514857924
ii= 204 , relres= 0.003684391232520088 , err= 0.00137421274299066
ii= 205 , relres= 0.003436926020005419 , err= 0.0013306328595027242
ii= 206 , relres= 0.003603947806789686 , err= 0.0012641771663374469
ii= 207 , relres= 0.0031600703014296157 , err= 0.0012169574420476392
ii= 208 , relres= 0.0031445595227096437 , err= 0.0011798471308273413
ii= 209 , relres= 0.003156719949262179 , err= 0.0011601556014716502
ii= 210 , relres= 0.0032124323207589484 , err= 0.0011284388827199445
ii= 211 , relres= 0.002840453055530792 , err= 0.001093465188680673
ii= 212 , relres= 0.0026762054838569338 , err= 0.0010764011025613924
ii= 213 , relres= 0.002920422145134889 , err= 0.0010235319605248156
ii= 214 , relres= 0.0026079261946609987 , err= 0.0009977827168194998
ii= 215 , relres= 0.0026541719245309703 , err= 0.0009845933824194442
ii= 216 , relres= 0.0024324871944608746 , err= 0.0009533125851053057
ii= 217 , relres= 0.002295590282315074 , err= 0.000893968240124789
ii= 218 , relres= 0.0023405475273130458 , err= 0.0008658187527220044
ii= 219 , relres= 0.0024334940715474247 , err= 0.0008476266165154975
ii= 220 , relres= 0.002144391143077441 , err= 0.0008087457244803375
ii= 221 , relres= 0.0020205938615974685 , err= 0.0008012249484546593
ii= 222 , relres= 0.002029518992864907 , err= 0.0007466119187008149
ii= 223 , relres= 0.0019178369572372194 , err= 0.0007196350454432633
ii= 224 , relres= 0.0018609007189773147 , err= 0.0006975616067656953
ii= 225 , relres= 0.0017328830319949446 , err= 0.0006593621645012143
ii= 226 , relres= 0.0016762770634731069 , err= 0.0006379506836678325
ii= 227 , relres= 0.0017988644602056159 , err= 0.0006178805386488803
ii= 228 , relres= 0.0018176562630366783 , err= 0.000585013357207978
ii= 229 , relres= 0.0016801051664094369 , err= 0.0005709375481031625
ii= 230 , relres= 0.001466824760699164 , err= 0.0005506038574735759
ii= 231 , relres= 0.0015516864544187659 , err= 0.0005285642007509795
ii= 232 , relres= 0.0015039603956202897 , err= 0.0005101873186800895
ii= 233 , relres= 0.0014698064969244583 , err= 0.00050083735574477
ii= 234 , relres= 0.001408315965778493 , err= 0.0004905080219136806
ii= 235 , relres= 0.0013800290564065878 , err= 0.0004661943620268417
ii= 236 , relres= 0.0013988395036672438 , err= 0.00044788957700725587
ii= 237 , relres= 0.001203221611554853 , err= 0.0004349726313624125
ii= 238 , relres= 0.0013795770864281846 , err= 0.0004264398739268195
ii= 239 , relres= 0.001257343155400186 , err= 0.00040349715631738656
ii= 240 , relres= 0.0011318579322708185 , err= 0.0003911828948660206
ii= 241 , relres= 0.0010860489927303298 , err= 0.0003839522012488136
ii= 242 , relres= 0.0011821815902064714 , err= 0.0003702057512332764
```

```
ii= 243 , relres= 0.0010089061359490196 , err= 0.00035216098159693716
ii= 244 , relres= 0.0009785746867845637 , err= 0.000347313652353857
ii= 245 , relres= 0.0010640217642738775 , err= 0.0003292864480316843
ii= 246 , relres= 0.000899031253170698 , err= 0.0003186264315746542
ii= 247 , relres= 0.000887473455901669 , err= 0.0003127622540138926
ii= 248 , relres= 0.0008459922030463775 , err= 0.0003001992539057049
ii= 249 , relres= 0.0008678380103184741 , err= 0.0002902651297778308
ii= 250 , relres= 0.0007835819071842699 , err= 0.00028099121631588856
ii= 251 , relres= 0.0007815406310618933 , err= 0.00027841423535830033
ii= 252 , relres= 0.0007309872558977504 , err= 0.0002597713022526462
ii= 253 , relres= 0.0006749630733668985 , err= 0.000253094189553386
ii= 254 , relres= 0.0007089965602579232 , err= 0.0002501608170795162
ii= 255 , relres= 0.0006851383427020576 , err= 0.00024204160937391428
ii= 256 , relres= 0.0006559480041616589 , err= 0.00023432434105594935
ii= 257 , relres= 0.0006109773843447858 , err= 0.0002259512833191499
ii= 258 , relres= 0.0006184083063487965 , err= 0.00022407006874599584
ii= 259 , relres= 0.0005956988989115555 , err= 0.0002091224636663743
ii= 260 , relres= 0.0005602817356345644 , err= 0.00020307795837162923
ii= 261 , relres= 0.0005328234549293341 , err= 0.0001958893675199069
ii= 262 , relres= 0.0005541243693300983 , err= 0.0001906494830891509
ii= 263 , relres= 0.0005480264654911114 , err= 0.00018384757267623207
ii= 264 , relres= 0.0004971580345474109 , err= 0.00017900221072964951
ii= 265 , relres= 0.000526625472923345 , err= 0.00017461165843525478
ii= 266 , relres= 0.0004639689541460289 , err= 0.00016557305224278556
ii= 267 , relres= 0.00045151187527374864 , err= 0.00016079115459211764
ii= 268 , relres= 0.0004397922181087401 , err= 0.00015520180591834965
ii= 269 , relres= 0.00041568771362594694 , err= 0.0001486654193190338
ii= 270 , relres= 0.0004155941717135306 , err= 0.00014670115590300267
ii= 271 , relres= 0.00037907736636876645 , err= 0.00014112092227473773
ii= 272 , relres= 0.0003931393074731857 , err= 0.00013384328150100369
ii= 273 , relres= 0.00037968466908747426 , err= 0.00013036492214082656
ii= 274 , relres= 0.00034703735506037484 , err= 0.0001282813601477319
ii= 275 , relres= 0.0003799385905690965 , err= 0.0001233751828718138
ii= 276 , relres= 0.0003242392708627704 , err= 0.00011896784160201094
ii= 277 , relres= 0.00032814082094431983 , err= 0.00011792035802482669
ii= 278 , relres= 0.0003268169330267596 , err= 0.00011393894114727604
ii= 279 , relres= 0.00032993454659225316 , err= 0.00011070179864878588
ii= 280 , relres= 0.00030991294617340517 , err= 0.0001047928595306744
ii= 281 , relres= 0.0002787090926302655 , err= 0.00010031371404368917
ii= 282 , relres= 0.0002970853374201729 , err= 9.509611420839023e-05
ii= 283 , relres= 0.000266555656983343 , err= 9.144804836681105e-05
ii= 284 , relres= 0.00024739668072719574 , err= 8.96947789678775e-05
ii= 285 , relres= 0.0002614490734177228 , err= 8.814052917308763e-05
ii= 286 , relres= 0.0002549597643351522 , err= 8.273744280783145e-05
ii= 287 , relres= 0.0002357692038177355 , err= 8.054441413693557e-05
ii= 288 , relres= 0.00022629569371644198 , err= 7.694192813333787e-05
ii= 289 , relres= 0.00023556730380666985 , err= 7.449133155589437e-05
ii= 290 , relres= 0.00022409931999340654 , err= 7.268969659154639e-05
ii= 291 , relres= 0.00021045065388713516 , err= 7.105129000614747e-05
ii= 292 , relres= 0.0002035625819102551 , err= 6.849015862459499e-05
ii= 293 , relres= 0.0002027124139488826 , err= 6.719374810891503e-05
ii= 294 , relres= 0.00021106468976162882 , err= 6.504903043294415e-05
ii= 295 , relres= 0.00019054462610153174 , err= 6.351173265957181e-05
ii= 296 , relres= 0.00016648777324259633 , err= 6.207635180867813e-05
ii= 297 , relres= 0.00018971633868406675 , err= 6.066021852588348e-05
ii= 298 , relres= 0.00015760146969425297 , err= 5.78033340135815e-05
ii= 299 , relres= 0.00015346597316076138 , err= 5.587527629652053e-05
ii= 300 , relres= 0.00014737443534572602 , err= 5.46943093883662e-05
ii= 301 , relres= 0.00014718588663934412 , err= 5.277976276885843e-05
ii= 302 , relres= 0.00013533873475978734 , err= 5.1209455996688445e-05
ii= 303 , relres= 0.00013849555164817653 , err= 4.961525410100528e-05
```

```
ii= 304 , relres= 0.0001229312898409285 , err= 4.583348651411907e-05
ii= 305 , relres= 0.00012531728405356182 , err= 4.476845442730298e-05
ii= 306 , relres= 0.0001312886431497087 , err= 4.365188664057224e-05
ii= 307 , relres= 0.00012490495682624314 , err= 4.105617162917662e-05
ii= 308 , relres= 0.0001178932809500321 , err= 4.005694515205715e-05
ii= 309 , relres= 0.00011491908024877956 , err= 3.906309718368895e-05
ii= 310 , relres= 0.00011133833129955402 , err= 3.819107477608954e-05
ii= 311 , relres= 0.00010924256933011789 , err= 3.7508666585710126e-05
ii= 312 , relres= 0.0001040597694282288 , err= 3.5327018105547656e-05
ii= 313 , relres= 9.995291773183788e-05 , err= 3.490321134623183e-05
ii= 314 , relres= 9.83864862207775e-05 , err= 3.414517660019619e-05
ii= 315 , relres= 9.742399157066674e-05 , err= 3.277439635600724e-05
ii= 316 , relres= 0.00010531681793087145 , err= 3.178986963575169e-05
ii= 317 , relres= 8.848220425480683e-05 , err= 3.071908033879707e-05
ii= 318 , relres= 8.647540820949714e-05 , err= 2.9802233071694076e-05
ii= 319 , relres= 8.422676356009347e-05 , err= 2.903814079778572e-05
ii= 320 , relres= 8.610550839848479e-05 , err= 2.8288488946449112e-05
ii= 321 , relres= 7.954452649355059e-05 , err= 2.712320358580535e-05
ii= 322 , relres= 7.243566800844024e-05 , err= 2.613356101964125e-05
ii= 323 , relres= 7.057749265017812e-05 , err= 2.575605963383146e-05
ii= 324 , relres= 6.82786110024297e-05 , err= 2.420617997286127e-05
ii= 325 , relres= 6.524863574697066e-05 , err= 2.3558546777214848e-05
ii= 326 , relres= 6.714302118002136e-05 , err= 2.28252274202486e-05
ii= 327 , relres= 5.706484961958128e-05 , err= 2.1294768165653296e-05
ii= 328 , relres= 5.9314415469712285e-05 , err= 2.1069289561554717e-05
ii= 329 , relres= 6.061476796576662e-05 , err= 2.0253311276931793e-05
ii= 330 , relres= 5.633070934656707e-05 , err= 1.986454376299672e-05
ii= 331 , relres= 5.569141727271442e-05 , err= 1.9604558395353692e-05
ii= 332 , relres= 5.599067947486981e-05 , err= 1.8827022899196016e-05
ii= 333 , relres= 5.2949662478586485e-05 , err= 1.857305669965692e-05
ii= 334 , relres= 5.174081206413893e-05 , err= 1.8052859819951694e-05
ii= 335 , relres= 5.341248066989729e-05 , err= 1.7342139749531802e-05
ii= 336 , relres= 4.57267173524838e-05 , err= 1.6640219561491028e-05
ii= 337 , relres= 4.668649219493337e-05 , err= 1.6487218807811818e-05
ii= 338 , relres= 4.6774197340698656e-05 , err= 1.6000737546258035e-05
ii= 339 , relres= 4.4935086782116184e-05 , err= 1.5068754213861797e-05
ii= 340 , relres= 3.841018643234958e-05 , err= 1.4478375953059722e-05
ii= 341 , relres= 4.4758772246454535e-05 , err= 1.3749343177158018e-05
ii= 342 , relres= 3.7685415806890095e-05 , err= 1.3144911015051691e-05
ii= 343 , relres= 3.998578891680538e-05 , err= 1.2865030983229672e-05
ii= 344 , relres= 3.583730802920671e-05 , err= 1.2193159570170035e-05
ii= 345 , relres= 3.676891522905124e-05 , err= 1.1616035285372311e-05
ii= 346 , relres= 3.4482338611543874e-05 , err= 1.1340331074990782e-05
ii= 347 , relres= 3.248859898358967e-05 , err= 1.099273377361903e-05
ii= 348 , relres= 3.230337567352932e-05 , err= 1.0844463181196426e-05
ii= 349 , relres= 3.072825900238667e-05 , err= 1.0572678588697398e-05
ii= 350 , relres= 2.862748005622616e-05 , err= 9.998096091449864e-06
ii= 351 , relres= 2.830774572533743e-05 , err= 9.909072939023522e-06
ii= 352 , relres= 2.9293204709291316e-05 , err= 9.566278535611775e-06
ii= 353 , relres= 2.9019807074903414e-05 , err= 9.173625932828404e-06
ii= 354 , relres= 2.7631292705239624e-05 , err= 8.945074647941076e-06
ii= 355 , relres= 2.6728228141873895e-05 , err= 8.735079173766071e-06
ii= 356 , relres= 2.5203899110931628e-05 , err= 8.366402420224147e-06
ii= 357 , relres= 2.4891059761584607e-05 , err= 8.284461496649288e-06
ii= 358 , relres= 2.5083041858673395e-05 , err= 7.824498969896967e-06
ii= 359 , relres= 2.2112210928156852e-05 , err= 7.5643956683098556e-06
ii= 360 , relres= 2.1333189258716958e-05 , err= 7.309017466199149e-06
ii= 361 , relres= 2.2536379385490114e-05 , err= 6.981025893658328e-06
ii= 362 , relres= 1.96212582848162e-05 , err= 6.736742182445203e-06
ii= 363 , relres= 1.8447626610028213e-05 , err= 6.510311342450965e-06
ii= 364 , relres= 1.900736681251031e-05 , err= 6.295676952669441e-06
```

```
ii= 365 , relres= 1.8504015848452133e-05 , err= 6.091476285394794e-06
ii= 366 , relres= 1.8555426570890753e-05 , err= 5.991727167900741e-06
ii= 367 , relres= 1.7502485322436643e-05 , err= 5.753763429390847e-06
ii= 368 , relres= 1.6194936147544417e-05 , err= 5.563249859788916e-06
ii= 369 , relres= 1.791577337175869e-05 , err= 5.296014074096077e-06
ii= 370 , relres= 1.681926729215779e-05 , err= 5.092369090919255e-06
ii= 371 , relres= 1.4585283006272578e-05 , err= 4.982490191433292e-06
ii= 372 , relres= 1.4631669404413488e-05 , err= 4.901018144629617e-06
ii= 373 , relres= 1.4935864608708996e-05 , err= 4.7141746034199945e-06
ii= 374 , relres= 1.3123965216035758e-05 , err= 4.589031869240549e-06
ii= 375 , relres= 1.3267370838411999e-05 , err= 4.494716000819859e-06
ii= 376 , relres= 1.2320076857650863e-05 , err= 4.3963650263019104e-06
ii= 377 , relres= 1.1610058829868856e-05 , err= 4.33615259129806e-06
ii= 378 , relres= 1.1984660330986132e-05 , err= 4.287138053156237e-06
ii= 379 , relres= 1.1799190542753444e-05 , err= 4.113369526423015e-06
ii= 380 , relres= 1.1633319736547975e-05 , err= 3.971544776126573e-06
ii= 381 , relres= 1.014580680269903e-05 , err= 3.89018309575458e-06
ii= 382 , relres= 1.0520030934142727e-05 , err= 3.7570051862312685e-06
ii= 383 , relres= 9.940553652818178e-06 , err= 3.6527331245841652e-06
ii= 384 , relres= 9.566393376695124e-06 , err= 3.498002101275379e-06
ii= 385 , relres= 9.138223659310286e-06 , err= 3.4377622313817063e-06
ii= 386 , relres= 1.03989492542893e-05 , err= 3.3373334231518425e-06
ii= 387 , relres= 9.93289212511214e-06 , err= 3.226764633695332e-06
ii= 388 , relres= 9.3253557247942e-06 , err= 3.097842226817874e-06
ii= 389 , relres= 8.075490256990464e-06 , err= 2.9589145950994517e-06
ii= 390 , relres= 8.074657647293867e-06 , err= 2.84595623648211e-06
ii= 391 , relres= 8.351285422933503e-06 , err= 2.7439349927979766e-06
ii= 392 , relres= 7.637078269529889e-06 , err= 2.6909077896019157e-06
ii= 393 , relres= 7.057892856470038e-06 , err= 2.5676553822134444e-06
ii= 394 , relres= 7.56000799974979e-06 , err= 2.497124347377304e-06
ii= 395 , relres= 7.693120765863271e-06 , err= 2.4148904129739904e-06
ii= 396 , relres= 6.743629226387022e-06 , err= 2.3374548524474174e-06
ii= 397 , relres= 6.1598293853911616e-06 , err= 2.2619534073779454e-06
ii= 398 , relres= 6.313366468021955e-06 , err= 2.225791496591058e-06
ii= 399 , relres= 6.234874903980258e-06 , err= 2.180993271159222e-06
ii= 400 , relres= 5.891369129400251e-06 , err= 2.0196057571768085e-06
ii= 401 , relres= 6.082518799140812e-06 , err= 1.9588899209456046e-06
ii= 402 , relres= 5.38449110759367e-06 , err= 1.9013794781546778e-06
ii= 403 , relres= 5.632833707605543e-06 , err= 1.8547212127699132e-06
ii= 404 , relres= 5.741222915610102e-06 , err= 1.7960815808282506e-06
ii= 405 , relres= 5.561056864968059e-06 , err= 1.7508071876264475e-06
ii= 406 , relres= 4.8900329896446995e-06 , err= 1.6724532864448022e-06
ii= 407 , relres= 4.7765951234590185e-06 , err= 1.6041039832553607e-06
ii= 408 , relres= 4.558611592862464e-06 , err= 1.5669317355580615e-06
ii= 409 , relres= 4.644166690564067e-06 , err= 1.5189164370302372e-06
ii= 410 , relres= 4.580288151391653e-06 , err= 1.4582817746705069e-06
ii= 411 , relres= 4.1247807708131425e-06 , err= 1.426796181448659e-06
ii= 412 , relres= 4.241173379720184e-06 , err= 1.4125889660282988e-06
ii= 413 , relres= 3.892451415535947e-06 , err= 1.3130152432095518e-06
ii= 414 , relres= 3.607708079925201e-06 , err= 1.2722947460762464e-06
ii= 415 , relres= 3.5131366363194934e-06 , err= 1.2609075424373673e-06
ii= 416 , relres= 3.6710344159386e-06 , err= 1.2372221442624143e-06
ii= 417 , relres= 3.3098148792856245e-06 , err= 1.1744048497664467e-06
ii= 418 , relres= 3.1134999331199195e-06 , err= 1.145224237938497e-06
ii= 419 , relres= 3.0068631632502883e-06 , err= 1.0976164040958224e-06
ii= 420 , relres= 2.9692391904261507e-06 , err= 1.0870721493075812e-06
ii= 421 , relres= 3.1099699546549144e-06 , err= 1.0406378851075424e-06
ii= 422 , relres= 2.803779005814151e-06 , err= 1.002083857791965e-06
ii= 423 , relres= 2.734075289944258e-06 , err= 9.790688555387554e-07
ii= 424 , relres= 2.692558345154163e-06 , err= 9.645478820654425e-07
ii= 425 , relres= 2.6481801589044666e-06 , err= 9.364760827070581e-07
```

```
ii= 426 , relres= 2.7066559068133255e-06 , err= 8.954083478273905e-07
ii= 427 , relres= 2.304281141804918e-06 , err= 8.642648422659659e-07
ii= 428 , relres= 2.2704987237282525e-06 , err= 8.46657785646525e-07
ii= 429 , relres= 2.224575113996069e-06 , err= 8.154141537973741e-07
ii= 430 , relres= 2.297920478782968e-06 , err= 7.942922643345711e-07
ii= 431 , relres= 2.3287585648676032e-06 , err= 7.672912317884779e-07
ii= 432 , relres= 2.178114953639321e-06 , err= 7.454261735790884e-07
ii= 433 , relres= 1.8503696264076128e-06 , err= 7.288633065015797e-07
ii= 434 , relres= 1.961717509289566e-06 , err= 7.080408103159333e-07
ii= 435 , relres= 1.948984284576653e-06 , err= 6.800203642993985e-07
ii= 436 , relres= 1.8169195645358925e-06 , err= 6.688242301244482e-07
ii= 437 , relres= 1.6665102311390079e-06 , err= 6.49742047310915e-07
ii= 438 , relres= 1.6614683567410351e-06 , err= 6.325012870333136e-07
ii= 439 , relres= 1.607471999142201e-06 , err= 6.095460728523743e-07
ii= 440 , relres= 1.53728586271817e-06 , err= 5.742263158239762e-07
ii= 441 , relres= 1.5377104370655324e-06 , err= 5.629590533927508e-07
ii= 442 , relres= 1.5481828459087948e-06 , err= 5.515055554165456e-07
ii= 443 , relres= 1.460708157077635e-06 , err= 5.37149632888033e-07
ii= 444 , relres= 1.4518871387258765e-06 , err= 5.257505859484403e-07
ii= 445 , relres= 1.586025441835225e-06 , err= 5.022652177281831e-07
ii= 446 , relres= 1.2833486818127463e-06 , err= 4.848082942040316e-07
ii= 447 , relres= 1.2291520929606303e-06 , err= 4.696171466196085e-07
ii= 448 , relres= 1.2059371173938043e-06 , err= 4.617269545855095e-07
ii= 449 , relres= 1.21628331437916e-06 , err= 4.410980399909826e-07
ii= 450 , relres= 1.264045304930428e-06 , err= 4.225884506679862e-07
ii= 451 , relres= 1.0728387161577141e-06 , err= 4.0244663251055913e-07
ii= 452 , relres= 1.0589712805898212e-06 , err= 3.98265728395698e-07
ii= 453 , relres= 1.0757713005332117e-06 , err= 3.9168872594030865e-07
ii= 454 , relres= 1.052657092653766e-06 , err= 3.649794285462121e-07
ii= 455 , relres= 1.0258222055311925e-06 , err= 3.543629470947045e-07
ii= 456 , relres= 9.916733592596929e-07 , err= 3.4923671043014355e-07
ii= 457 , relres= 9.128654233382788e-07 , err= 3.31066053684399e-07
ii= 458 , relres= 9.208807929844687e-07 , err= 3.209120261593966e-07
ii= 459 , relres= 9.404595579911647e-07 , err= 3.1221715618567654e-07
ii= 460 , relres= 8.118735088408285e-07 , err= 2.921572508411968e-07
ii= 461 , relres= 7.75139291924443e-07 , err= 2.879229448616165e-07
ii= 462 , relres= 7.934631473080601e-07 , err= 2.800317602108104e-07
ii= 463 , relres= 7.620590933072588e-07 , err= 2.5788522938131384e-07
ii= 464 , relres= 6.930691966042477e-07 , err= 2.5457792359289736e-07
ii= 465 , relres= 7.51078232847471e-07 , err= 2.4733845782475175e-07
ii= 466 , relres= 6.849970094676358e-07 , err= 2.3826585624710273e-07
ii= 467 , relres= 6.809677490628516e-07 , err= 2.3285880530345097e-07
ii= 468 , relres= 6.221625901530464e-07 , err= 2.1548521711544463e-07
ii= 469 , relres= 5.853789683014853e-07 , err= 2.0677109939443623e-07
ii= 470 , relres= 5.640676239932542e-07 , err= 2.0365540128303831e-07
ii= 471 , relres= 6.414006600808326e-07 , err= 1.9325272657187255e-07
ii= 472 , relres= 5.60606138620124e-07 , err= 1.875496362756725e-07
ii= 473 , relres= 4.850199828276333e-07 , err= 1.7840346059707401e-07
ii= 474 , relres= 5.379434286149686e-07 , err= 1.7530576286072484e-07
ii= 475 , relres= 5.140841864450677e-07 , err= 1.661786696888873e-07
ii= 476 , relres= 4.782351086890638e-07 , err= 1.6242334422223207e-07
ii= 477 , relres= 4.867541901002345e-07 , err= 1.536146148957772e-07
ii= 478 , relres= 4.3499448296331295e-07 , err= 1.4790263994582765e-07
ii= 479 , relres= 4.0925292517685784e-07 , err= 1.4378292118606923e-07
ii= 480 , relres= 3.9476376264296996e-07 , err= 1.3619383147489165e-07
ii= 481 , relres= 3.768071282512256e-07 , err= 1.3120711993677638e-07
ii= 482 , relres= 3.7998903647583876e-07 , err= 1.2953743764895093e-07
ii= 483 , relres= 3.5559855670802355e-07 , err= 1.2703576669199185e-07
ii= 484 , relres= 3.6870139204591547e-07 , err= 1.2408369881549106e-07
ii= 485 , relres= 3.7120255103428454e-07 , err= 1.1895556917018934e-07
ii= 486 , relres= 3.2827927672577673e-07 , err= 1.1512441397325677e-07
```

```

ii= 487 , relres= 3.259440791230323e-07 , err= 1.1379601024881509e-07
ii= 488 , relres= 3.2867965137772816e-07 , err= 1.1023130406077397e-07
ii= 489 , relres= 2.989535741793554e-07 , err= 1.0732750626738148e-07
ii= 490 , relres= 3.114386976670507e-07 , err= 1.0273013339665718e-07
ii= 491 , relres= 3.0093432444185357e-07 , err= 9.749664158000364e-08
ii= 492 , relres= 2.7441888429676066e-07 , err= 9.58736096181994e-08
ii= 493 , relres= 2.5423964256092386e-07 , err= 9.425503403990745e-08
ii= 494 , relres= 2.850068023255222e-07 , err= 9.054943090555151e-08
ii= 495 , relres= 2.8311172567974363e-07 , err= 8.762989003608997e-08
ii= 496 , relres= 2.604821415300303e-07 , err= 8.388987124814774e-08
ii= 497 , relres= 2.364985392920235e-07 , err= 8.103854033789203e-08
ii= 498 , relres= 2.2282378077950677e-07 , err= 7.884813961999389e-08
ii= 499 , relres= 2.3385944948494102e-07 , err= 7.617242831313157e-08
ii= 500 , relres= 2.2010100105052943e-07 , err= 7.239725884618558e-08
ii= 501 , relres= 2.0773393459798175e-07 , err= 7.003862638844121e-08
ii= 502 , relres= 2.1922544774767856e-07 , err= 6.817673716377411e-08
ii= 503 , relres= 1.9042061484630612e-07 , err= 6.623913734873836e-08
ii= 504 , relres= 1.8345586901892348e-07 , err= 6.503209378634099e-08
ii= 505 , relres= 1.8548447594350703e-07 , err= 6.320444634168186e-08
ii= 506 , relres= 1.7245781765745971e-07 , err= 5.883617818846374e-08
ii= 507 , relres= 1.620542141197705e-07 , err= 5.780857249789875e-08
ii= 508 , relres= 1.5737880594040457e-07 , err= 5.696555609238093e-08
ii= 509 , relres= 1.6917635190374231e-07 , err= 5.576964309870469e-08
ii= 510 , relres= 1.6543454894080826e-07 , err= 5.383697829991437e-08
ii= 511 , relres= 1.4435069526126337e-07 , err= 5.072702946634911e-08
ii= 512 , relres= 1.4281720958158663e-07 , err= 5.0221811568140144e-08
ii= 513 , relres= 1.403548609335828e-07 , err= 4.904530747041634e-08
ii= 514 , relres= 1.42182791017851e-07 , err= 4.707678020544068e-08
ii= 515 , relres= 1.3940612154633656e-07 , err= 4.639312140363903e-08
ii= 516 , relres= 1.3043497327542938e-07 , err= 4.482128142718842e-08
ii= 517 , relres= 1.2249989631712878e-07 , err= 4.41989987551414e-08
ii= 518 , relres= 1.2620715562453454e-07 , err= 4.292345650612485e-08
ii= 519 , relres= 1.1540117653201899e-07 , err= 4.189714223350343e-08
ii= 520 , relres= 1.1887399794472413e-07 , err= 4.102949473525048e-08
ii= 521 , relres= 1.0946298919915045e-07 , err= 3.861674016001191e-08
ii= 522 , relres= 1.0044329937563512e-07 , err= 3.709235414800905e-08
ii= 523 , relres= 9.448359268328851e-08 , err= 3.683090559113713e-08
ii= 524 , relres= 9.879413558667564e-08 , err= 3.619350744965802e-08
ii= 525 , relres= 9.437572419476308e-08 , err= 3.426152098890454e-08
ii= 526 , relres= 8.996237243890236e-08 , err= 3.307283937662759e-08
ii= 527 , relres= 8.310028608920757e-08 , err= 3.142736983118778e-08
ii= 528 , relres= 8.536662823262602e-08 , err= 3.094998631419764e-08
ii= 529 , relres= 8.538928491185346e-08 , err= 2.9398764029448068e-08
ii= 530 , relres= 8.230053509392614e-08 , err= 2.8784471197535415e-08
ii= 531 , relres= 7.514674954922986e-08 , err= 2.7985108138029462e-08
ii= 532 , relres= 7.629561371043827e-08 , err= 2.759715315401793e-08
ii= 533 , relres= 6.979946442100137e-08 , err= 2.6905730061233318e-08
ii= 534 , relres= 8.77398538696452e-08 , err= 2.5445646222558752e-08
ii= 535 , relres= 6.685303053499081e-08 , err= 2.41168329631138e-08
ii= 536 , relres= 6.514741346615173e-08 , err= 2.3595481229515497e-08
ii= 537 , relres= 6.417688258907902e-08 , err= 2.2741489780699162e-08
ii= 538 , relres= 5.865288777202908e-08 , err= 2.2108931112493242e-08
ii= 539 , relres= 6.466179156740463e-08 , err= 2.1494957033143054e-08
ii= 540 , relres= 5.201741322536406e-08 , err= 1.9457716768915704e-08
ii= 541 , relres= 4.940363604561661e-08 , err= 1.9315382356004626e-08
ii= 542 , relres= 5.367874571552615e-08 , err= 1.8742656258065668e-08
ii= 543 , relres= 5.433124483611136e-08 , err= 1.77127791360098e-08
ii= 544 , relres= 5.172765976990939e-08 , err= 1.7079076201471844e-08
ii= 545 , relres= 4.502231233656795e-08 , err= 1.6359957614066953e-08
ii= 546 , relres= 4.468170551723223e-08 , err= 1.607596858655982e-08
ii= 547 , relres= 4.063157454018424e-08 , err= 1.528973668872357e-08

```

```
ii= 548 , relres= 4.425080908781778e-08 , err= 1.4603642941752239e-08
ii= 549 , relres= 4.099580359823164e-08 , err= 1.4004973419927956e-08
ii= 550 , relres= 4.033821156791083e-08 , err= 1.3663386807713905e-08
ii= 551 , relres= 3.7074875344241697e-08 , err= 1.3406524460392494e-08
ii= 552 , relres= 3.6859509549267646e-08 , err= 1.2906583923305047e-08
ii= 553 , relres= 3.734363218144417e-08 , err= 1.2689950981391352e-08
ii= 554 , relres= 3.3861533171245395e-08 , err= 1.1933033562191787e-08
ii= 555 , relres= 3.146159422453538e-08 , err= 1.181448490187476e-08
ii= 556 , relres= 3.076833367204892e-08 , err= 1.1349070336659567e-08
ii= 557 , relres= 3.254634516945629e-08 , err= 1.0989286361499282e-08
ii= 558 , relres= 3.0212583566242223e-08 , err= 1.019456006989883e-08
ii= 559 , relres= 2.95210883196112e-08 , err= 1.0038276496384331e-08
ii= 560 , relres= 2.9618089456528987e-08 , err= 9.843177999511136e-09
ii= 561 , relres= 2.701828012455457e-08 , err= 9.375357721580516e-09
ii= 562 , relres= 2.5921622487378173e-08 , err= 9.180862813265384e-09
ii= 563 , relres= 2.5312886169908198e-08 , err= 8.924702603564353e-09
ii= 564 , relres= 2.2768311804248842e-08 , err= 8.190637180612581e-09
ii= 565 , relres= 2.01760044873115e-08 , err= 8.04308921344499e-09
ii= 566 , relres= 2.1156947976761213e-08 , err= 7.89591669220823e-09
ii= 567 , relres= 2.22270894266603e-08 , err= 7.666039550859913e-09
ii= 568 , relres= 2.06156671807551e-08 , err= 7.125549679152811e-09
ii= 569 , relres= 2.0056226673844813e-08 , err= 7.0470324226876995e-09
ii= 570 , relres= 1.9301249685179855e-08 , err= 6.8541951855851005e-09
ii= 571 , relres= 1.8746271659673896e-08 , err= 6.7077476133111775e-09
ii= 572 , relres= 1.815618912909278e-08 , err= 6.571574237607934e-09
ii= 573 , relres= 1.857162784634898e-08 , err= 6.464702852389344e-09
ii= 574 , relres= 1.823448580891713e-08 , err= 6.272002054067755e-09
ii= 575 , relres= 1.760269207916984e-08 , err= 6.013758426736009e-09
ii= 576 , relres= 1.596605402896733e-08 , err= 5.858893661458449e-09
ii= 577 , relres= 1.5144324932788668e-08 , err= 5.770176849227243e-09
ii= 578 , relres= 1.565336144902325e-08 , err= 5.661523770681642e-09
ii= 579 , relres= 1.600409208225144e-08 , err= 5.466800229606482e-09
ii= 580 , relres= 1.5733404066062875e-08 , err= 5.352290990099391e-09
ii= 581 , relres= 1.504813967354795e-08 , err= 5.2124474134252e-09
ii= 582 , relres= 1.4190635522958208e-08 , err= 4.890852727174704e-09
ii= 583 , relres= 1.3362645476090814e-08 , err= 4.7440282319980155e-09
ii= 584 , relres= 1.2299420866099688e-08 , err= 4.546553586618148e-09
ii= 585 , relres= 1.2460135453264493e-08 , err= 4.391528181622326e-09
ii= 586 , relres= 1.188591746443674e-08 , err= 4.1821560250140335e-09
ii= 587 , relres= 1.1300963021947452e-08 , err= 3.99916672772318e-09
ii= 588 , relres= 1.022966468116623e-08 , err= 3.929102753573809e-09
ii= 589 , relres= 1.0552594263213369e-08 , err= 3.822149826770542e-09
ii= 590 , relres= 1.111665131201718e-08 , err= 3.751705428796463e-09
ii= 591 , relres= 1.032184998520516e-08 , err= 3.5560024655069946e-09
ii= 592 , relres= 9.670913329254658e-09 , err= 3.3747337630906584e-09
ii= 593 , relres= 9.362395847614402e-09 , err= 3.333409923523305e-09
ii= 594 , relres= 9.389601320755001e-09 , err= 3.2645924598665634e-09
ii= 595 , relres= 9.36060192561649e-09 , err= 3.150826124054962e-09
ii= 596 , relres= 9.56728303052271e-09 , err= 3.067552904388554e-09
ii= 597 , relres= 8.46282855788859e-09 , err= 2.9593311093157832e-09
ii= 598 , relres= 9.021520591722059e-09 , err= 2.8889950242855533e-09
ii= 599 , relres= 7.977872604659271e-09 , err= 2.734051215772297e-09
ii= 600 , relres= 7.509674473962223e-09 , err= 2.682728897966762e-09
ii= 601 , relres= 7.555945609131441e-09 , err= 2.6076716932522788e-09
ii= 602 , relres= 7.549020103696617e-09 , err= 2.500271254354771e-09
ii= 603 , relres= 6.888265317525219e-09 , err= 2.415998497648589e-09
ii= 604 , relres= 7.09145662000413e-09 , err= 2.342089611623371e-09
ii= 605 , relres= 6.162589647433699e-09 , err= 2.270438878325033e-09
ii= 606 , relres= 6.448219294955814e-09 , err= 2.2129151923790464e-09
ii= 607 , relres= 5.742762161012967e-09 , err= 2.143170022805621e-09
ii= 608 , relres= 5.736943172053687e-09 , err= 2.0578982791697144e-09
```

```
ii= 609 , relres= 5.601910241584167e-09 , err= 1.958576204056224e-09
ii= 610 , relres= 5.09693284401301e-09 , err= 1.9295569023811754e-09
ii= 611 , relres= 5.137434529292532e-09 , err= 1.909283970459441e-09
ii= 612 , relres= 5.088316318320614e-09 , err= 1.8555427026726698e-09
ii= 613 , relres= 5.524407559513812e-09 , err= 1.7674139875747143e-09
ii= 614 , relres= 4.715195425435017e-09 , err= 1.7170110504978084e-09
ii= 615 , relres= 4.692555925664592e-09 , err= 1.6765134507228333e-09
ii= 616 , relres= 4.593364910424318e-09 , err= 1.6347505750117596e-09
ii= 617 , relres= 4.378519498195384e-09 , err= 1.6143043280850657e-09
ii= 618 , relres= 4.527716775492831e-09 , err= 1.550744337849981e-09
ii= 619 , relres= 4.0941075312745695e-09 , err= 1.4741631773924485e-09
ii= 620 , relres= 4.172973248523998e-09 , err= 1.4402678606463757e-09
ii= 621 , relres= 3.6822837158116444e-09 , err= 1.383057261944051e-09
ii= 622 , relres= 3.575243668193574e-09 , err= 1.35873324384694e-09
ii= 623 , relres= 3.305356469324484e-09 , err= 1.3009374100983955e-09
ii= 624 , relres= 3.5753480195363706e-09 , err= 1.2394862183311955e-09
ii= 625 , relres= 3.0279870199018757e-09 , err= 1.1772540396136753e-09
ii= 626 , relres= 3.0205740082868e-09 , err= 1.157086598767657e-09
ii= 627 , relres= 2.8998260738112255e-09 , err= 1.1202582840998349e-09
ii= 628 , relres= 3.0609846661722253e-09 , err= 1.094029503910605e-09
ii= 629 , relres= 3.493950338862186e-09 , err= 1.0500307013488242e-09
ii= 630 , relres= 2.7580755881278848e-09 , err= 9.578317422835672e-10
ii= 631 , relres= 2.5714470588819157e-09 , err= 9.435871821947222e-10
ii= 632 , relres= 2.7096551215107214e-09 , err= 9.295063899792939e-10
ii= 633 , relres= 2.516170357193799e-09 , err= 8.909697597517754e-10
ii= 634 , relres= 2.4946677532273102e-09 , err= 8.734020604750802e-10
ii= 635 , relres= 2.3966742062964544e-09 , err= 8.449490773460638e-10
ii= 636 , relres= 2.6910238691015236e-09 , err= 8.16795210925812e-10
ii= 637 , relres= 2.225629682888562e-09 , err= 7.721123709325503e-10
ii= 638 , relres= 2.035973474552266e-09 , err= 7.547329615706879e-10
ii= 639 , relres= 1.9681210858362256e-09 , err= 7.263969334398814e-10
ii= 640 , relres= 2.07053377868135e-09 , err= 7.108238730027376e-10
ii= 641 , relres= 2.0851493798309997e-09 , err= 6.876454186013003e-10
ii= 642 , relres= 1.8236700840279538e-09 , err= 6.51379105137314e-10
ii= 643 , relres= 1.6855916695140268e-09 , err= 6.327893344579589e-10
ii= 644 , relres= 1.7002020971314876e-09 , err= 6.23855462680588e-10
ii= 645 , relres= 1.6377720921233687e-09 , err= 5.948235565366638e-10
ii= 646 , relres= 1.6492308139015423e-09 , err= 5.81738172574355e-10
ii= 647 , relres= 1.527436641800854e-09 , err= 5.559609507341755e-10
ii= 648 , relres= 1.5453273158840452e-09 , err= 5.38389244375834e-10
ii= 649 , relres= 1.3888889512066514e-09 , err= 5.303436255133109e-10
ii= 650 , relres= 1.3645358605537011e-09 , err= 5.211093463550285e-10
ii= 651 , relres= 1.3163943383355889e-09 , err= 5.075200592878146e-10
ii= 652 , relres= 1.3789037817192396e-09 , err= 4.986501093452563e-10
ii= 653 , relres= 1.3919920818016894e-09 , err= 4.881082417333008e-10
ii= 654 , relres= 1.2525352673154758e-09 , err= 4.4364186228225753e-10
ii= 655 , relres= 1.1757209475472695e-09 , err= 4.3528217275268504e-10
ii= 656 , relres= 1.2547353310512759e-09 , err= 4.265890462257898e-10
ii= 657 , relres= 1.1246388854308594e-09 , err= 4.140623563137773e-10
ii= 658 , relres= 1.122766825649023e-09 , err= 4.0361681930382055e-10
ii= 659 , relres= 1.0643294094792059e-09 , err= 3.876924853778736e-10
ii= 660 , relres= 1.0642637398841735e-09 , err= 3.7685255228113855e-10
ii= 661 , relres= 1.0468636635130857e-09 , err= 3.6368775857224274e-10
ii= 662 , relres= 9.914584345556634e-10 , err= 3.530475093707353e-10
ii= 663 , relres= 9.392998368494597e-10 , err= 3.4756791641109355e-10
ii= 664 , relres= 9.478363033054966e-10 , err= 3.3713257655708613e-10
ii= 665 , relres= 9.168023180643636e-10 , err= 3.301148269420272e-10
ii= 666 , relres= 8.968586072142256e-10 , err= 3.182067686767127e-10
ii= 667 , relres= 8.057717025900293e-10 , err= 3.021485823135433e-10
ii= 668 , relres= 8.502261669897489e-10 , err= 2.979840010802762e-10
ii= 669 , relres= 8.133846907771484e-10 , err= 2.888043811177566e-10
```

```

ii= 670 , relres= 7.49034427154389e-10 , err= 2.6253539242286686e-10
ii= 671 , relres= 7.829761315127147e-10 , err= 2.5625258931992175e-10
ii= 672 , relres= 7.731831508688057e-10 , err= 2.498110877115214e-10
ii= 673 , relres= 7.103749094825806e-10 , err= 2.4087394479352405e-10
ii= 674 , relres= 6.885569476087889e-10 , err= 2.350410011952269e-10
ii= 675 , relres= 7.012528977429288e-10 , err= 2.1756793646678772e-10
ii= 676 , relres= 6.580384659012827e-10 , err= 2.1404795655808023e-10
ii= 677 , relres= 6.110197287623397e-10 , err= 2.0405367973893538e-10
ii= 678 , relres= 5.508771859873595e-10 , err= 2.0051880090345515e-10
ii= 679 , relres= 5.625011280922943e-10 , err= 1.952007847108748e-10
ii= 680 , relres= 5.671888332861756e-10 , err= 1.8635410227250223e-10
ii= 681 , relres= 5.288400538452216e-10 , err= 1.7851855006878746e-10
ii= 682 , relres= 5.180105280715943e-10 , err= 1.696088876435596e-10
ii= 683 , relres= 4.921987695254501e-10 , err= 1.6727659728449953e-10
ii= 684 , relres= 5.02781762563318e-10 , err= 1.630047532158257e-10
ii= 685 , relres= 4.474514427627699e-10 , err= 1.561489371804909e-10
ii= 686 , relres= 4.532587391155926e-10 , err= 1.5482789776660573e-10
ii= 687 , relres= 4.780600153472362e-10 , err= 1.4541024676052481e-10
ii= 688 , relres= 4.0322632381339445e-10 , err= 1.4119056217239561e-10
ii= 689 , relres= 3.966479432362868e-10 , err= 1.3729270094277345e-10
ii= 690 , relres= 3.878311205249933e-10 , err= 1.3447796330285303e-10
ii= 691 , relres= 4.0406440283707405e-10 , err= 1.3075281349491405e-10
ii= 692 , relres= 4.2222322477028637e-10 , err= 1.2795740841994072e-10
ii= 693 , relres= 3.769183946324569e-10 , err= 1.2466227072273556e-10
ii= 694 , relres= 3.552320552306335e-10 , err= 1.2031954680594338e-10
ii= 695 , relres= 3.345629900276668e-10 , err= 1.1574011724786878e-10
ii= 696 , relres= 3.164762487205996e-10 , err= 1.1227822179478468e-10
ii= 697 , relres= 3.0143296698484747e-10 , err= 1.1138195265631093e-10
ii= 698 , relres= 3.1287029008426404e-10 , err= 1.0813666633774313e-10
ii= 699 , relres= 3.059371473789276e-10 , err= 1.0310823896040518e-10
ii= 700 , relres= 2.9108029175645355e-10 , err= 1.0028007473865478e-10
ii= 701 , relres= 2.884736512107864e-10 , err= 9.718478496420407e-11
ii= 702 , relres= 2.5458992999288655e-10 , err= 9.561341791092333e-11
ii= 703 , relres= 2.6326466728200295e-10 , err= 9.319356387015878e-11
ii= 704 , relres= 2.632708116168653e-10 , err= 9.142783863424402e-11
ii= 705 , relres= 2.5488002456166935e-10 , err= 8.972366615619156e-11
ii= 706 , relres= 2.540050943220083e-10 , err= 8.775008895315198e-11
ii= 707 , relres= 2.4920288808461085e-10 , err= 8.487144145787045e-11
ii= 708 , relres= 2.2559386028524367e-10 , err= 8.209443843981193e-11
ii= 709 , relres= 2.121426325885769e-10 , err= 8.050785113462968e-11
ii= 710 , relres= 2.2458519840326175e-10 , err= 7.715355252316113e-11
ii= 711 , relres= 2.4459281745371704e-10 , err= 7.45146465616426e-11
ii= 712 , relres= 2.310516365425321e-10 , err= 7.296393927264801e-11
ii= 713 , relres= 2.0047498462343575e-10 , err= 7.092906805417524e-11
ii= 714 , relres= 1.986088569942439e-10 , err= 6.890039009364942e-11
ii= 715 , relres= 1.918409365677052e-10 , err= 6.752480359723697e-11
ii= 716 , relres= 1.9675517747036824e-10 , err= 6.628855586659403e-11
ii= 717 , relres= 1.9226493943452886e-10 , err= 6.265215776103947e-11
ii= 718 , relres= 1.7476602010417247e-10 , err= 6.177417607469319e-11
ii= 719 , relres= 2.0098575562695933e-10 , err= 5.939853601142148e-11
ii= 720 , relres= 1.9832370489264814e-10 , err= 5.753303647330009e-11
ii= 721 , relres= 1.6220048271228582e-10 , err= 5.5281129610164196e-11
ii= 722 , relres= 1.8059848678121118e-10 , err= 5.453186395409297e-11
ii= 723 , relres= 1.7487157470499903e-10 , err= 5.3134757010575266e-11
ii= 724 , relres= 1.6331804946944616e-10 , err= 5.267646077155813e-11
ii= 725 , relres= 1.591218740237017e-10 , err= 4.8407441973556555e-11
ii= 726 , relres= 1.6673736229284618e-10 , err= 4.678364687357252e-11
ii= 727 , relres= 1.572032310591943e-10 , err= 4.5010581847524074e-11
ii= 728 , relres= 1.420688650645984e-10 , err= 4.3620988993761187e-11
ii= 729 , relres= 1.3955944619875608e-10 , err= 4.2295756759996736e-11
ii= 730 , relres= 1.5635912828365334e-10 , err= 4.100882213847963e-11

```

```
ii= 731 , relres= 1.3179664135372574e-10 , err= 3.9447989517332067e-11
ii= 732 , relres= 1.453337316455164e-10 , err= 3.838041802554482e-11
ii= 733 , relres= 1.287474003850763e-10 , err= 3.785424680984008e-11
ii= 734 , relres= 1.3178327297985578e-10 , err= 3.586372603199879e-11
ii= 735 , relres= 1.3498787377543472e-10 , err= 3.512783701433889e-11
ii= 736 , relres= 1.3260463657310617e-10 , err= 3.411800874426025e-11
ii= 737 , relres= 1.3751844147149086e-10 , err= 3.31057030374962e-11
ii= 738 , relres= 1.26429829746067e-10 , err= 3.232031247326546e-11
ii= 739 , relres= 1.140331848230814e-10 , err= 3.185577051016661e-11
ii= 740 , relres= 1.0420581913269019e-10 , err= 3.137556996537476e-11
ii= 741 , relres= 1.0157109248058121e-10 , err= 3.095609011876014e-11
ii= 742 , relres= 1.0525687174140492e-10 , err= 3.0421515518546846e-11
ii= 743 , relres= 1.1034016989414417e-10 , err= 2.9910585873362576e-11
ii= 744 , relres= 1.0792927097482923e-10 , err= 2.9398913918000536e-11
ii= 745 , relres= 1.0467483792718772e-10 , err= 2.8932856355551215e-11
ii= 746 , relres= 1.2173176209881344e-10 , err= 2.8433036366643916e-11
ii= 747 , relres= 1.0968856880562898e-10 , err= 2.7909677407265804e-11
ii= 748 , relres= 1.0807794990232615e-10 , err= 2.746948986418014e-11
ii= 749 , relres= 1.0512666953830247e-10 , err= 2.7192246203396826e-11
ii= 750 , relres= 1.2827011786757463e-10 , err= 2.6971315367047204e-11
ii= 751 , relres= 1.2443429685721368e-10 , err= 2.672779938311848e-11
ii= 752 , relres= 1.2484144578038016e-10 , err= 2.6486899959562566e-11
ii= 753 , relres= 1.2337376156246573e-10 , err= 2.626584530670763e-11
ii= 754 , relres= 1.1578566579182782e-10 , err= 2.6116739327853883e-11
ii= 755 , relres= 1.2828225333247886e-10 , err= 2.5989477085721678e-11
ii= 756 , relres= 1.4249511505380192e-10 , err= 2.5813227722832206e-11
ii= 757 , relres= 1.3236908523761416e-10 , err= 2.5598911527688894e-11
ii= 758 , relres= 1.3807750963985934e-10 , err= 2.541699317779158e-11
ii= 759 , relres= 1.5117755605302117e-10 , err= 2.5244479373088305e-11
ii= 760 , relres= 1.3559682067223857e-10 , err= 2.5058029556158302e-11
ii= 761 , relres= 1.2472808572548384e-10 , err= 2.4902003629642165e-11
ii= 762 , relres= 1.347084597940143e-10 , err= 2.4777528491346252e-11
ii= 763 , relres= 1.4523829711577957e-10 , err= 2.4664026316792225e-11
ii= 764 , relres= 1.392071001278739e-10 , err= 2.458778216620503e-11
ii= 765 , relres= 1.3037257525541885e-10 , err= 2.4548595792846764e-11
ii= 766 , relres= 1.3414682087033198e-10 , err= 2.4515695445549763e-11
ii= 767 , relres= 1.2937117569848214e-10 , err= 2.4488358885359543e-11
ii= 768 , relres= 1.4744083997958268e-10 , err= 2.44842908144347e-11
ii= 769 , relres= 1.2326319640096128e-10 , err= 2.448499851841639e-11
ii= 770 , relres= 1.280130666196832e-10 , err= 2.447932532041331e-11
ii= 771 , relres= 1.2233569788638635e-10 , err= 2.447010186781364e-11
ii= 772 , relres= 1.2005635997567754e-10 , err= 2.4459331290326165e-11
ii= 773 , relres= 1.1950197916681365e-10 , err= 2.4443775499527825e-11
ii= 774 , relres= 1.232897631394907e-10 , err= 2.4430849412489014e-11
ii= 775 , relres= 1.2653404221139631e-10 , err= 2.4436847625555397e-11
ii= 776 , relres= 1.4142702676228423e-10 , err= 2.4455376604399075e-11
ii= 777 , relres= 1.2729198065232498e-10 , err= 2.4484327847290826e-11
ii= 778 , relres= 1.3319077837580043e-10 , err= 2.4525017697242895e-11
ii= 779 , relres= 1.1909484299982868e-10 , err= 2.4562753764767977e-11
ii= 780 , relres= 1.1442370908698499e-10 , err= 2.459111782964696e-11
ii= 781 , relres= 1.117163534227284e-10 , err= 2.4618180326354185e-11
ii= 782 , relres= 1.058714744976859e-10 , err= 2.46516947806237e-11
ii= 783 , relres= 1.280251723923397e-10 , err= 2.4687946967981052e-11
ii= 784 , relres= 1.3458699738786462e-10 , err= 2.4724131963422713e-11
ii= 785 , relres= 1.0608484843225465e-10 , err= 2.4761993000236684e-11
ii= 786 , relres= 1.4789316708465265e-10 , err= 2.4802479559021196e-11
ii= 787 , relres= 1.2787570468757504e-10 , err= 2.4850560162635974e-11
ii= 788 , relres= 1.2813760195481444e-10 , err= 2.4911400402632297e-11
ii= 789 , relres= 1.0866274253664415e-10 , err= 2.4972343715102e-11
ii= 790 , relres= 1.3289854498362468e-10 , err= 2.5022737220103515e-11
ii= 791 , relres= 1.181710017591479e-10 , err= 2.5063079699088756e-11
```

```
ii= 792 , relres= 1.2518988024704384e-10 , err= 2.5101982614659327e-11
ii= 793 , relres= 1.1730530930613557e-10 , err= 2.515100126177148e-11
ii= 794 , relres= 1.0857536590789504e-10 , err= 2.5214420295926257e-11
ii= 795 , relres= 1.2339863233155976e-10 , err= 2.5276530544443822e-11
ii= 796 , relres= 1.0532078550820983e-10 , err= 2.5326512506183768e-11
ii= 797 , relres= 1.051486990616792e-10 , err= 2.5372524026259154e-11
ii= 798 , relres= 1.13106166361906e-10 , err= 2.5423781164483052e-11
ii= 799 , relres= 1.0720946073048552e-10 , err= 2.5478629627636152e-11
ii= 800 , relres= 1.238196395841846e-10 , err= 2.5533507756151812e-11
ii= 801 , relres= 1.1672774083661027e-10 , err= 2.559115515620344e-11
ii= 802 , relres= 1.1386539840208524e-10 , err= 2.5648500527855534e-11
ii= 803 , relres= 1.0283165298584682e-10 , err= 2.5696773466424847e-11
ii= 804 , relres= 9.868506328432605e-11 , err= 2.5732473087505877e-11
GN: none
ii= 0 , relres= 1.6440777647452876 , err= 0.9357225259849997
ii= 1 , relres= 1.4112489183456107 , err= 0.8893443917417876
ii= 2 , relres= 1.2148104403162934 , err= 0.8444414106357909
ii= 3 , relres= 1.138556331141885 , err= 0.8074844298742812
ii= 4 , relres= 1.0586264770237819 , err= 0.7636870187818043
ii= 5 , relres= 0.9696556251431689 , err= 0.7298871505554937
ii= 6 , relres= 0.8549784321491137 , err= 0.6804906725839434
ii= 7 , relres= 0.8088523993948487 , err= 0.6322779321208837
ii= 8 , relres= 0.7599498187946571 , err= 0.590871729365155
ii= 9 , relres= 0.741464374782051 , err= 0.5525501058316237
ii= 10 , relres= 0.5993965319349124 , err= 0.5167328909148183
ii= 11 , relres= 0.5687215345527542 , err= 0.47984445092060896
ii= 12 , relres= 0.537903669857156 , err= 0.4454733037803047
ii= 13 , relres= 0.48439725304008835 , err= 0.41492070584228347
ii= 14 , relres= 0.45958690462338586 , err= 0.38425429772177
ii= 15 , relres= 0.411439961770369 , err= 0.35154615460068556
ii= 16 , relres= 0.3922902912014336 , err= 0.32065258483754905
ii= 17 , relres= 0.33812661948017 , err= 0.2931490639069771
ii= 18 , relres= 0.3399114237896448 , err= 0.26797564974748583
ii= 19 , relres= 0.29989244892266825 , err= 0.24590141767551504
ii= 20 , relres= 0.2893525496378512 , err= 0.22512718501318368
ii= 21 , relres= 0.25939760740992435 , err= 0.20970231430301253
ii= 22 , relres= 0.21941228034628965 , err= 0.19464182747191183
ii= 23 , relres= 0.1997388715093074 , err= 0.18139366125216885
ii= 24 , relres= 0.17358320357425377 , err= 0.1697956433197558
ii= 25 , relres= 0.15816076867902767 , err= 0.158552564199308
ii= 26 , relres= 0.15419958976512252 , err= 0.14835179599478052
ii= 27 , relres= 0.13324905645352722 , err= 0.13918949217504992
ii= 28 , relres= 0.15154863309074748 , err= 0.13226674205403519
ii= 29 , relres= 0.12044674812171127 , err= 0.12653481834137248
ii= 30 , relres= 0.11063319507267981 , err= 0.11981320003220161
ii= 31 , relres= 0.10294241251285237 , err= 0.11206487827974726
ii= 32 , relres= 0.09209138602436147 , err= 0.10591341838941258
ii= 33 , relres= 0.08049908732250849 , err= 0.09965839457307274
ii= 34 , relres= 0.07662369009770839 , err= 0.09380594278001567
ii= 35 , relres= 0.071876670782493 , err= 0.08900920190394936
ii= 36 , relres= 0.06560778114368647 , err= 0.08395120909036775
ii= 37 , relres= 0.06331773047129838 , err= 0.07857252516048328
ii= 38 , relres= 0.057290428593686146 , err= 0.0747868596460464
ii= 39 , relres= 0.05948274704338275 , err= 0.07108845961739978
ii= 40 , relres= 0.053104988935592005 , err= 0.06808435480430619
ii= 41 , relres= 0.046421999102366385 , err= 0.06423993417365846
ii= 42 , relres= 0.04510373231977359 , err= 0.06097148293882916
ii= 43 , relres= 0.043858928706780806 , err= 0.057209536905480994
ii= 44 , relres= 0.0393997825248406 , err= 0.05365231654617648
ii= 45 , relres= 0.043512005121383035 , err= 0.05078153390071018
ii= 46 , relres= 0.03697014804619865 , err= 0.048116197618371895
```

```
ii= 47 , relres= 0.030874297790833744 , err= 0.04465841656009271
ii= 48 , relres= 0.030042628898117922 , err= 0.04151574418466084
ii= 49 , relres= 0.02970308658989317 , err= 0.03874354324608426
ii= 50 , relres= 0.028521524895672955 , err= 0.03619877002564949
ii= 51 , relres= 0.028202909825603337 , err= 0.033907699204699694
ii= 52 , relres= 0.030464784724618606 , err= 0.03215812290723135
ii= 53 , relres= 0.023225700582385957 , err= 0.030139785755632142
ii= 54 , relres= 0.02316388338024898 , err= 0.027792983844632344
ii= 55 , relres= 0.021664290939247433 , err= 0.025631195007002363
ii= 56 , relres= 0.01858750735398126 , err= 0.02327689704134387
ii= 57 , relres= 0.018665087139380893 , err= 0.021321685246867096
ii= 58 , relres= 0.01774684300747036 , err= 0.019833090461571112
ii= 59 , relres= 0.016223261266910392 , err= 0.018025148487905304
ii= 60 , relres= 0.017651157446394068 , err= 0.01655789494898393
ii= 61 , relres= 0.014324176044041365 , err= 0.015320397283771606
ii= 62 , relres= 0.013211510168437792 , err= 0.014101185107330055
ii= 63 , relres= 0.012370266954909986 , err= 0.012775090941961172
ii= 64 , relres= 0.011034018458133276 , err= 0.01172516381872153
ii= 65 , relres= 0.009848149531157995 , err= 0.01067141697320808
ii= 66 , relres= 0.009052792605090442 , err= 0.009725047008988639
ii= 67 , relres= 0.008752872259471887 , err= 0.00880626316238902
ii= 68 , relres= 0.0089756526712966 , err= 0.008267198974488249
ii= 69 , relres= 0.007000657771176559 , err= 0.007675733842528624
ii= 70 , relres= 0.006664141554251126 , err= 0.007011275509764598
ii= 71 , relres= 0.007454707788028623 , err= 0.0065652598538372695
ii= 72 , relres= 0.005759922031431697 , err= 0.006202012829611976
ii= 73 , relres= 0.0062452803706389325 , err= 0.005691932645573089
ii= 74 , relres= 0.005807284428373403 , err= 0.00539630948300627
ii= 75 , relres= 0.00501438709796543 , err= 0.005091097382698512
ii= 76 , relres= 0.0042096108814991835 , err= 0.004699969658429286
ii= 77 , relres= 0.0036592833769764016 , err= 0.004298347968930057
ii= 78 , relres= 0.004119543921357681 , err= 0.0039875069154203715
ii= 79 , relres= 0.0036882598082230845 , err= 0.0037861224390039927
ii= 80 , relres= 0.0030384105393117633 , err= 0.003541027041573164
ii= 81 , relres= 0.0028042588548952657 , err= 0.0032542417265639116
ii= 82 , relres= 0.002462212081050531 , err= 0.003013877568924812
ii= 83 , relres= 0.002222658906909016 , err= 0.002771722326533355
ii= 84 , relres= 0.0020194034105176007 , err= 0.0025454814634622404
ii= 85 , relres= 0.0022300953434869764 , err= 0.0023649197976254774
ii= 86 , relres= 0.0018908665503359168 , err= 0.0022255672378490928
ii= 87 , relres= 0.0018335780079859515 , err= 0.0020691483713030775
ii= 88 , relres= 0.0015297322328407595 , err= 0.0019159056460226758
ii= 89 , relres= 0.0013333443712835577 , err= 0.001767060470764973
ii= 90 , relres= 0.0012760987541061732 , err= 0.0016078588297422445
ii= 91 , relres= 0.0012127328607323263 , err= 0.0014682997609989114
ii= 92 , relres= 0.001519913473730202 , err= 0.0013829490256611204
ii= 93 , relres= 0.0011966885192014223 , err= 0.0012947357788486778
ii= 94 , relres= 0.0013156330972854582 , err= 0.0012138117687104668
ii= 95 , relres= 0.0009814053057464647 , err= 0.0011318775959632578
ii= 96 , relres= 0.0009236406066707129 , err= 0.001034489274878379
ii= 97 , relres= 0.0009275583391361968 , err= 0.0009698774060150627
ii= 98 , relres= 0.0007583430630166338 , err= 0.0008952779263145276
ii= 99 , relres= 0.0007338783629525415 , err= 0.0008019408351103337
ii= 100 , relres= 0.000809694740386175 , err= 0.0007299707643140483
ii= 101 , relres= 0.0007359559490322989 , err= 0.0006843290883498096
ii= 102 , relres= 0.0006515850148356095 , err= 0.000620563946800302
ii= 103 , relres= 0.0006368551045981398 , err= 0.000570557165950081
ii= 104 , relres= 0.0005030267427728243 , err= 0.0005214119750579751
ii= 105 , relres= 0.000476119383493945 , err= 0.0004689873348203647
ii= 106 , relres= 0.000507981661257896 , err= 0.0004359861957956443
ii= 107 , relres= 0.0004211010577261039 , err= 0.00039792552056246784
```

```
ii= 108 , relres= 0.0004499614829612536 , err= 0.0003694199839562718
ii= 109 , relres= 0.00036725763378902794 , err= 0.00033923633276400247
ii= 110 , relres= 0.0003498485751951289 , err= 0.00030771116284213226
ii= 111 , relres= 0.0003590440824651918 , err= 0.00028659334604293186
ii= 112 , relres= 0.0002477808210441839 , err= 0.0002603518007548542
ii= 113 , relres= 0.00023502548036403767 , err= 0.00023219205076986644
ii= 114 , relres= 0.0002599919605616193 , err= 0.00021076264070057213
ii= 115 , relres= 0.0002018235034081361 , err= 0.00019483426667022964
ii= 116 , relres= 0.00019515827047540774 , err= 0.00017618912571064195
ii= 117 , relres= 0.00022966961146733803 , err= 0.00016558476323878915
ii= 118 , relres= 0.00017160514192702655 , err= 0.0001530486233035825
ii= 119 , relres= 0.00017550469272468056 , err= 0.00014131878078781513
ii= 120 , relres= 0.00013229236900430755 , err= 0.00012837437870991174
ii= 121 , relres= 0.00012720721394767493 , err= 0.0001140098598296458
ii= 122 , relres= 0.00012782684179884636 , err= 0.0001029104173349081
ii= 123 , relres= 0.00010618813091957684 , err= 9.468733750543233e-05
ii= 124 , relres= 8.834933019078912e-05 , err= 8.575837434721763e-05
ii= 125 , relres= 0.00010072797546573216 , err= 7.83240952955866e-05
ii= 126 , relres= 7.678131129452614e-05 , err= 7.22916699570581e-05
ii= 127 , relres= 6.832416358751416e-05 , err= 6.544751376303194e-05
ii= 128 , relres= 6.563118888248906e-05 , err= 5.9758490680231824e-05
ii= 129 , relres= 6.362980845437058e-05 , err= 5.6089735989156994e-05
ii= 130 , relres= 6.274624720197071e-05 , err= 5.152727673924375e-05
ii= 131 , relres= 5.5076479752390164e-05 , err= 4.8666617726226166e-05
ii= 132 , relres= 4.750339045243914e-05 , err= 4.4469917041479006e-05
ii= 133 , relres= 4.746428335770398e-05 , err= 4.1368340216740354e-05
ii= 134 , relres= 3.6416596045688193e-05 , err= 3.7634538000209364e-05
ii= 135 , relres= 3.437312203111783e-05 , err= 3.37016532621679e-05
ii= 136 , relres= 4.0478001934392006e-05 , err= 3.1594780517377356e-05
ii= 137 , relres= 3.037546943107984e-05 , err= 2.887495731939134e-05
ii= 138 , relres= 3.7347461252797696e-05 , err= 2.6548666076227703e-05
ii= 139 , relres= 3.118787552988664e-05 , err= 2.4658641811181543e-05
ii= 140 , relres= 2.7678265186014825e-05 , err= 2.2856858492595273e-05
ii= 141 , relres= 2.5968786974241642e-05 , err= 2.078720214471535e-05
ii= 142 , relres= 2.424452411701089e-05 , err= 1.9404569057696007e-05
ii= 143 , relres= 2.145703417575545e-05 , err= 1.7653296514716418e-05
ii= 144 , relres= 1.9408160055968145e-05 , err= 1.6246094102714746e-05
ii= 145 , relres= 1.5705611446696063e-05 , err= 1.4782967216287597e-05
ii= 146 , relres= 1.657854705431329e-05 , err= 1.3394332488061222e-05
ii= 147 , relres= 1.4039415393792816e-05 , err= 1.2193037279394942e-05
ii= 148 , relres= 1.1956266154496387e-05 , err= 1.0929200228816613e-05
ii= 149 , relres= 1.0865758075086168e-05 , err= 9.743461605902033e-06
ii= 150 , relres= 1.180479387016031e-05 , err= 8.96799345484749e-06
ii= 151 , relres= 1.0537774967453468e-05 , err= 8.099760789275101e-06
ii= 152 , relres= 9.700252404599209e-06 , err= 7.498752168459719e-06
ii= 153 , relres= 7.79158652504353e-06 , err= 6.876154266609793e-06
ii= 154 , relres= 6.9998271714466615e-06 , err= 6.108812487387168e-06
ii= 155 , relres= 6.844856395536646e-06 , err= 5.517292482341743e-06
ii= 156 , relres= 5.879197019220106e-06 , err= 5.125129888200472e-06
ii= 157 , relres= 5.347981917278854e-06 , err= 4.682638334680228e-06
ii= 158 , relres= 5.187879416010268e-06 , err= 4.405471872884896e-06
ii= 159 , relres= 5.191047359401262e-06 , err= 4.043641007045493e-06
ii= 160 , relres= 4.766089277488865e-06 , err= 3.852946010788586e-06
ii= 161 , relres= 3.549455881300017e-06 , err= 3.5670998435453203e-06
ii= 162 , relres= 3.813650239657935e-06 , err= 3.255814561439177e-06
ii= 163 , relres= 3.6024228282696913e-06 , err= 3.032394517362828e-06
ii= 164 , relres= 3.329459205579464e-06 , err= 2.872055389777375e-06
ii= 165 , relres= 2.5097027039149087e-06 , err= 2.6349941588252726e-06
ii= 166 , relres= 2.159373158167042e-06 , err= 2.3879725435786767e-06
ii= 167 , relres= 2.0319856319903694e-06 , err= 2.175341971138098e-06
ii= 168 , relres= 1.9596112283697412e-06 , err= 1.990174343029024e-06
```

```
ii= 169 , relres= 2.1938980069602423e-06 , err= 1.8808098870577097e-06
ii= 170 , relres= 1.7035379534137273e-06 , err= 1.7700959341275952e-06
ii= 171 , relres= 1.4628133172968164e-06 , err= 1.6689625425548001e-06
ii= 172 , relres= 1.5036668911402299e-06 , err= 1.5470142962892429e-06
ii= 173 , relres= 1.369423676571385e-06 , err= 1.4653630295649553e-06
ii= 174 , relres= 1.1620003327974955e-06 , err= 1.3563769656028454e-06
ii= 175 , relres= 1.1906735262974822e-06 , err= 1.2769673148346182e-06
ii= 176 , relres= 9.928134233336615e-07 , err= 1.1875304948120356e-06
ii= 177 , relres= 1.1438399736714805e-06 , err= 1.126144517595932e-06
ii= 178 , relres= 8.380084359379033e-07 , err= 1.0604227986438557e-06
ii= 179 , relres= 7.970064410971822e-07 , err= 1.0017032595266785e-06
ii= 180 , relres= 8.735985755925807e-07 , err= 9.412607448302585e-07
ii= 181 , relres= 7.758976407315916e-07 , err= 9.089041967550857e-07
ii= 182 , relres= 6.242917021726148e-07 , err= 8.470039357336138e-07
ii= 183 , relres= 5.789599635630501e-07 , err= 7.846531538789234e-07
ii= 184 , relres= 6.034342003328031e-07 , err= 7.487248725967691e-07
ii= 185 , relres= 5.208282660749324e-07 , err= 6.997285359412831e-07
ii= 186 , relres= 4.663062056542958e-07 , err= 6.453246311732799e-07
ii= 187 , relres= 4.299635176815519e-07 , err= 5.961757983527134e-07
ii= 188 , relres= 4.6944779919797364e-07 , err= 5.533645140994455e-07
ii= 189 , relres= 4.801073018299076e-07 , err= 5.166710430744367e-07
ii= 190 , relres= 4.0969410489953207e-07 , err= 4.895940069789034e-07
ii= 191 , relres= 3.9596072307429515e-07 , err= 4.4958524269120693e-07
ii= 192 , relres= 3.546577135629295e-07 , err= 4.2472643099666893e-07
ii= 193 , relres= 3.2191823895504566e-07 , err= 3.87352435571228e-07
ii= 194 , relres= 3.588525761280149e-07 , err= 3.641285503073306e-07
ii= 195 , relres= 2.6092382924338116e-07 , err= 3.3721652080738707e-07
ii= 196 , relres= 2.8028083952403503e-07 , err= 3.0711458163895173e-07
ii= 197 , relres= 2.4257883526116567e-07 , err= 2.894698032422476e-07
ii= 198 , relres= 1.984672995349003e-07 , err= 2.622471564776104e-07
ii= 199 , relres= 2.024751880802782e-07 , err= 2.4031759615013335e-07
ii= 200 , relres= 2.2743046527150658e-07 , err= 2.258882663851093e-07
ii= 201 , relres= 2.0558277229758524e-07 , err= 2.09965373081932e-07
ii= 202 , relres= 1.5823122772557807e-07 , err= 1.9738897785145628e-07
ii= 203 , relres= 2.085506748922599e-07 , err= 1.8449494552660938e-07
ii= 204 , relres= 1.617784335652362e-07 , err= 1.734622749602269e-07
ii= 205 , relres= 1.4987473702701896e-07 , err= 1.6336135166785e-07
ii= 206 , relres= 1.2025992697704124e-07 , err= 1.5120639085362217e-07
ii= 207 , relres= 1.1666284236981506e-07 , err= 1.386932518590566e-07
ii= 208 , relres= 9.906845966644648e-08 , err= 1.2778797309013994e-07
ii= 209 , relres= 9.230241841884429e-08 , err= 1.1633158479628028e-07
ii= 210 , relres= 1.1043588400126692e-07 , err= 1.0788565171941393e-07
ii= 211 , relres= 7.809561427572667e-08 , err= 1.0011622602104867e-07
ii= 212 , relres= 7.554792307374116e-08 , err= 9.00036819270014e-08
ii= 213 , relres= 8.447718271733027e-08 , err= 8.264594439903455e-08
ii= 214 , relres= 7.253858244247511e-08 , err= 7.689393155600557e-08
ii= 215 , relres= 6.831347529867756e-08 , err= 7.026400613137765e-08
ii= 216 , relres= 6.534155503862273e-08 , err= 6.623624910925354e-08
ii= 217 , relres= 5.661555904096513e-08 , err= 6.105987302562109e-08
ii= 218 , relres= 5.737429306189359e-08 , err= 5.6717544223884154e-08
ii= 219 , relres= 5.544203424983311e-08 , err= 5.277206641839823e-08
ii= 220 , relres= 4.524247629927564e-08 , err= 5.001081661923364e-08
ii= 221 , relres= 4.1606845014669377e-08 , err= 4.638080744032526e-08
ii= 222 , relres= 3.8940922689168405e-08 , err= 4.3415358516254535e-08
ii= 223 , relres= 3.0909066655360856e-08 , err= 3.973040803565348e-08
ii= 224 , relres= 3.7366007227132974e-08 , err= 3.6498955817729264e-08
ii= 225 , relres= 2.8823128067578106e-08 , err= 3.4391734601570524e-08
ii= 226 , relres= 2.868004618466438e-08 , err= 3.162416831420235e-08
ii= 227 , relres= 3.1912595684158725e-08 , err= 3.0242042651714456e-08
ii= 228 , relres= 2.4692853434390128e-08 , err= 2.798553749765198e-08
ii= 229 , relres= 2.412496495554391e-08 , err= 2.5678911691260977e-08
```

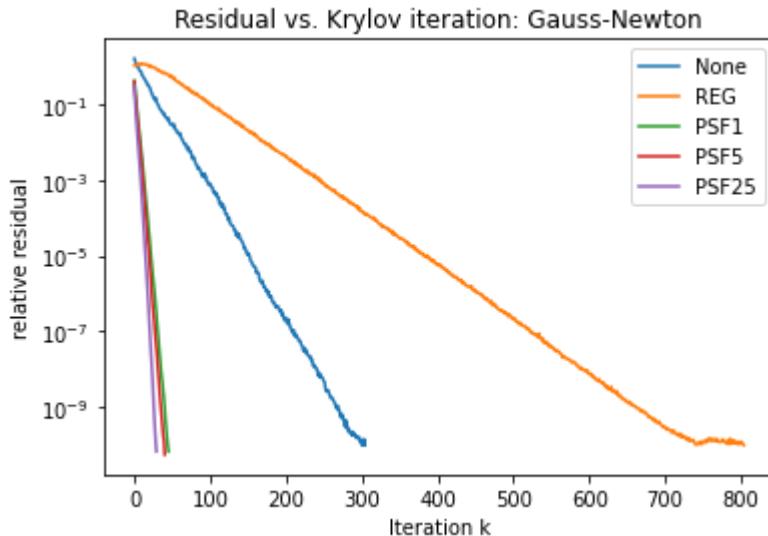
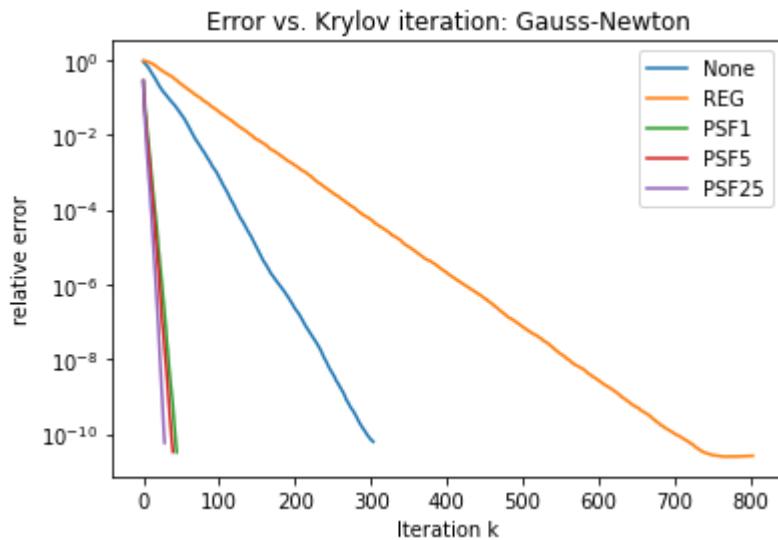
```
ii= 230 , relres= 2.268292562707147e-08 , err= 2.349809560953038e-08
ii= 231 , relres= 2.1142279256192955e-08 , err= 2.1689516581461482e-08
ii= 232 , relres= 2.2143708610304976e-08 , err= 2.0038834981399542e-08
ii= 233 , relres= 1.7720624294312443e-08 , err= 1.8773694073536424e-08
ii= 234 , relres= 1.8791465954244146e-08 , err= 1.6882161082244947e-08
ii= 235 , relres= 1.4985703273091734e-08 , err= 1.5687214733016117e-08
ii= 236 , relres= 1.2052141250818937e-08 , err= 1.3846873220764972e-08
ii= 237 , relres= 1.300558089506123e-08 , err= 1.2195253417898247e-08
ii= 238 , relres= 1.2483358594706235e-08 , err= 1.1262103275943748e-08
ii= 239 , relres= 1.0742287088801988e-08 , err= 1.0111864493650859e-08
ii= 240 , relres= 1.0648120702407947e-08 , err= 9.478565265160468e-09
ii= 241 , relres= 1.0112741667531548e-08 , err= 8.672160648130228e-09
ii= 242 , relres= 7.468106175442573e-09 , err= 7.84853658679782e-09
ii= 243 , relres= 6.759784445704707e-09 , err= 7.0030251303804385e-09
ii= 244 , relres= 7.937920166764453e-09 , err= 6.390753377945574e-09
ii= 245 , relres= 6.1661386473582175e-09 , err= 5.809552842397312e-09
ii= 246 , relres= 6.9927240599008174e-09 , err= 5.362191121989172e-09
ii= 247 , relres= 6.22164191390659e-09 , err= 4.973280083938261e-09
ii= 248 , relres= 5.999264602028605e-09 , err= 4.59614875311062e-09
ii= 249 , relres= 5.356426914946894e-09 , err= 4.258008745569757e-09
ii= 250 , relres= 4.838234554290878e-09 , err= 3.9895287381074796e-09
ii= 251 , relres= 4.4800246079234736e-09 , err= 3.730984062912046e-09
ii= 252 , relres= 3.1064005643306528e-09 , err= 3.3528109390694344e-09
ii= 253 , relres= 2.6300532618518325e-09 , err= 2.99238252665692e-09
ii= 254 , relres= 2.805380905896008e-09 , err= 2.679758407017144e-09
ii= 255 , relres= 2.844596354062239e-09 , err= 2.5085774843860036e-09
ii= 256 , relres= 2.7131259125809382e-09 , err= 2.3372721303874447e-09
ii= 257 , relres= 2.1976733885012897e-09 , err= 2.146153051327812e-09
ii= 258 , relres= 2.0518604254556956e-09 , err= 1.9796797807882847e-09
ii= 259 , relres= 1.735185454147977e-09 , err= 1.825408404811435e-09
ii= 260 , relres= 1.7855939710499512e-09 , err= 1.648506453546533e-09
ii= 261 , relres= 1.7725217874878623e-09 , err= 1.5569492073651393e-09
ii= 262 , relres= 1.592274021733428e-09 , err= 1.422357537517578e-09
ii= 263 , relres= 1.4406871889529294e-09 , err= 1.331100944263868e-09
ii= 264 , relres= 1.3328747140188476e-09 , err= 1.206072613721825e-09
ii= 265 , relres= 1.1748396869505744e-09 , err= 1.117322110585183e-09
ii= 266 , relres= 9.798878237561682e-10 , err= 9.810071886211718e-10
ii= 267 , relres= 9.283898543769787e-10 , err= 8.515370429359724e-10
ii= 268 , relres= 1.1028773532441453e-09 , err= 7.813530930940099e-10
ii= 269 , relres= 9.61164322940208e-10 , err= 7.309478192970206e-10
ii= 270 , relres= 7.773488060102567e-10 , err= 6.772815099897709e-10
ii= 271 , relres= 7.719718613036959e-10 , err= 6.226961845995265e-10
ii= 272 , relres= 5.513847944999339e-10 , err= 5.696726010281568e-10
ii= 273 , relres= 6.269862919644067e-10 , err= 5.152383763956936e-10
ii= 274 , relres= 5.108071121070668e-10 , err= 4.825805193093827e-10
ii= 275 , relres= 6.14805552422748e-10 , err= 4.5057255129290306e-10
ii= 276 , relres= 4.574669079324446e-10 , err= 4.158075234410963e-10
ii= 277 , relres= 4.216643807164939e-10 , err= 3.964797601077258e-10
ii= 278 , relres= 3.0376184143317057e-10 , err= 3.5995715546751344e-10
ii= 279 , relres= 4.5055630153500305e-10 , err= 3.3031894209362126e-10
ii= 280 , relres= 2.786180649141526e-10 , err= 3.1089448295397943e-10
ii= 281 , relres= 3.0163336743413645e-10 , err= 2.8008101451478283e-10
ii= 282 , relres= 2.9050240401319896e-10 , err= 2.608227450406999e-10
ii= 283 , relres= 2.1824935473718308e-10 , err= 2.3427054243798296e-10
ii= 284 , relres= 2.4113684548952343e-10 , err= 2.1139664157256629e-10
ii= 285 , relres= 2.1371225652990998e-10 , err= 1.9257175386953056e-10
ii= 286 , relres= 2.1670661600280508e-10 , err= 1.8032454941723245e-10
ii= 287 , relres= 1.7850222605099536e-10 , err= 1.6620772081016806e-10
ii= 288 , relres= 2.2255294175054306e-10 , err= 1.5178135794439985e-10
ii= 289 , relres= 1.8766410217510282e-10 , err= 1.4367934982444515e-10
ii= 290 , relres= 2.0245684517921584e-10 , err= 1.3461980994783448e-10
```

```

ii= 291 , relres= 1.9159702376119632e-10 , err= 1.248267680509958e-10
ii= 292 , relres= 1.5776422112484193e-10 , err= 1.170860495623102e-10
ii= 293 , relres= 1.7005294420454958e-10 , err= 1.0974958457170467e-10
ii= 294 , relres= 1.4889539704571703e-10 , err= 9.978112336186869e-11
ii= 295 , relres= 1.5913753289455817e-10 , err= 9.363222190123434e-11
ii= 296 , relres= 1.5972247450841867e-10 , err= 8.663141541015906e-11
ii= 297 , relres= 1.591474203276274e-10 , err= 8.175625184578287e-11
ii= 298 , relres= 1.1959802196007064e-10 , err= 7.716887074113213e-11
ii= 299 , relres= 1.3119974839705128e-10 , err= 7.309589564137527e-11
ii= 300 , relres= 1.0017851238731534e-10 , err= 6.927649000081253e-11
ii= 301 , relres= 1.0794832286816241e-10 , err= 6.633721392919584e-11
ii= 302 , relres= 1.0131954370702663e-10 , err= 6.418693232986968e-11
ii= 303 , relres= 1.4332408394771788e-10 , err= 6.164304865398364e-11
ii= 304 , relres= 9.840750678386585e-11 , err= 5.914857126302248e-11

```

Out[48]:



In [51]:

```

print('FULL: get truth:')
x_true, _, _, _ = cg_wrapper(H_linop, b, all_p_linop[-1])

all_relres = list()
all_errs = list()
for nb, p_linop in zip(all_num_batches, all_p_linop):
    print('FULL: psf'+str(nb))
    _, _, relres, errs = cg_wrapper(H_linop, b, p_linop, x_true=x_true)

```

```

    all_relres.append(relres)
    all_errs.append(errs)

    print('FULL: reg')
    _, _, all_relres_reg, all_err_reg = cg_wrapper(H_linop, b, ireg_linop, x_true=x_
    _true)

    print('FULL: none')
    _, _, all_relres_none, all_err_none = cg_wrapper(H_linop, b, None, x_true=x_true)

    plt.semilogy(all_err_none[:-1])
    plt.semilogy(all_err_reg[:-1])
    for errs in all_errs:
        plt.semilogy(errs[:-1])
    plt.title('Error vs. Krylov iteration: Full Hessian')
    plt.xlabel('Iteration k')
    plt.ylabel('relative error')
    plt.legend(['None', 'REG'] + ['PSF'+str(nb) for nb in all_num_batches])

    plt.figure()
    plt.semilogy(all_relres_none[:-1])
    plt.semilogy(all_relres_reg[:-1])
    for relres in all_relres:
        plt.semilogy(relres)
    plt.title('Residual vs. Krylov iteration: Full Hessian')
    plt.xlabel('Iteration k')
    plt.ylabel('relative residual')
    plt.legend(['None', 'REG'] + ['PSF'+str(nb) for nb in all_num_batches])

```

```

FULL: get truth:
ii= 0 , relres= 0.319674396935505 , err= nan
ii= 1 , relres= 0.12338139967969793 , err= nan
ii= 2 , relres= 0.05948674177893956 , err= nan
ii= 3 , relres= 0.02873594496854482 , err= nan
ii= 4 , relres= 0.01418554528833904 , err= nan
ii= 5 , relres= 0.00775870120358675 , err= nan
ii= 6 , relres= 0.0044343581089372665 , err= nan
ii= 7 , relres= 0.00248020041969897 , err= nan
ii= 8 , relres= 0.001324608253049718 , err= nan
ii= 9 , relres= 0.000617542597586057 , err= nan
ii= 10 , relres= 0.0003216338959501857 , err= nan
ii= 11 , relres= 0.0001549030005628696 , err= nan
ii= 12 , relres= 7.330714805743457e-05 , err= nan
ii= 13 , relres= 3.583398698171643e-05 , err= nan
ii= 14 , relres= 1.5106764496808158e-05 , err= nan
ii= 15 , relres= 7.500704649516748e-06 , err= nan
ii= 16 , relres= 3.656191923008437e-06 , err= nan
ii= 17 , relres= 1.650325380189637e-06 , err= nan
ii= 18 , relres= 7.623676552210472e-07 , err= nan
ii= 19 , relres= 3.245544044215603e-07 , err= nan
ii= 20 , relres= 1.4506954553005417e-07 , err= nan
ii= 21 , relres= 6.083777063152004e-08 , err= nan
ii= 22 , relres= 2.4245697067888834e-08 , err= nan
ii= 23 , relres= 1.0608085407559839e-08 , err= nan
ii= 24 , relres= 4.478103933021794e-09 , err= nan
ii= 25 , relres= 1.7783985867929981e-09 , err= nan
ii= 26 , relres= 7.064984678714521e-10 , err= nan
ii= 27 , relres= 2.7483752261740746e-10 , err= nan
ii= 28 , relres= 1.553410117278497e-10 , err= nan
ii= 29 , relres= 1.217151932143965e-10 , err= nan
ii= 30 , relres= 8.943094301368401e-11 , err= nan

```

```

ii= 0 , relres= 0.4300209164059238 , err= 0.27691797964492504
ii= 1 , relres= 0.2550433461925696 , err= 0.12336711236007922
ii= 2 , relres= 0.14696653897776804 , err= 0.06953950936656078
ii= 3 , relres= 0.09295347967567866 , err= 0.03692585890389791
ii= 4 , relres= 0.05725169420270372 , err= 0.024790491947334032
ii= 5 , relres= 0.0352595950417005 , err= 0.013785929762935365
ii= 6 , relres= 0.023387714399200222 , err= 0.009051319468285003
ii= 7 , relres= 0.014958574529111653 , err= 0.005652499956739436
ii= 8 , relres= 0.009734061573384464 , err= 0.0034864937402672297
ii= 9 , relres= 0.006130470700688983 , err= 0.002141342800626258
ii= 10 , relres= 0.0038034018240823923 , err= 0.0012831150825472935
ii= 11 , relres= 0.0022717755030508056 , err= 0.0007321378240081803
ii= 12 , relres= 0.001415992348541561 , err= 0.0004609009899673127
ii= 13 , relres= 0.0008883928411219346 , err= 0.00028026423262406217
ii= 14 , relres= 0.0005338721391208557 , err= 0.0001838964382855129
ii= 15 , relres= 0.0003409903134344588 , err= 0.00010798224353826696
ii= 16 , relres= 0.00021212758030883123 , err= 7.73592843288748e-05
ii= 17 , relres= 0.00013220470959055412 , err= 4.5744475317468166e-05
ii= 18 , relres= 8.240845676758521e-05 , err= 2.8975678489522947e-05
ii= 19 , relres= 4.7528738134394125e-05 , err= 1.7164313217716693e-05
ii= 20 , relres= 2.9516858432421074e-05 , err= 9.300076563052609e-06
ii= 21 , relres= 1.6809052536773595e-05 , err= 5.896084079521181e-06
ii= 22 , relres= 9.522070122733246e-06 , err= 3.444055252378933e-06
ii= 23 , relres= 5.693809101798435e-06 , err= 2.105207359360449e-06
ii= 24 , relres= 3.4943140594402585e-06 , err= 1.3804577578385691e-06
ii= 25 , relres= 2.1524337811788366e-06 , err= 7.831765472164672e-07
ii= 26 , relres= 1.3910866750423546e-06 , err= 4.6658010759590473e-07
ii= 27 , relres= 8.515498601786927e-07 , err= 2.9964699671829037e-07
ii= 28 , relres= 5.114188721165706e-07 , err= 1.6347060151316396e-07
ii= 29 , relres= 2.9503074228594875e-07 , err= 1.053104784199235e-07
ii= 30 , relres= 1.5790408982932484e-07 , err= 5.720611906193069e-08
ii= 31 , relres= 9.35709792162145e-08 , err= 3.032089890253132e-08
ii= 32 , relres= 5.476815920585457e-08 , err= 1.8032331623511446e-08
ii= 33 , relres= 3.093488656359108e-08 , err= 1.0515879011516325e-08
ii= 34 , relres= 1.7711272909706417e-08 , err= 5.776944821265922e-09
ii= 35 , relres= 1.0096841561973975e-08 , err= 3.6088870318093692e-09
ii= 36 , relres= 6.237550537590045e-09 , err= 2.1129206045773635e-09
ii= 37 , relres= 4.195656164612939e-09 , err= 1.326288064549208e-09
ii= 38 , relres= 2.5726425126182753e-09 , err= 8.640585252187048e-10
ii= 39 , relres= 1.7120070422327036e-09 , err= 5.154498693737611e-10
ii= 40 , relres= 9.173380041887532e-10 , err= 3.0523426435630295e-10
ii= 41 , relres= 4.677123555138939e-10 , err= 1.64847271122453e-10
ii= 42 , relres= 2.58841706571471e-10 , err= 9.067836654751276e-11
ii= 43 , relres= 1.6349688828841616e-10 , err= 5.368696666237465e-11
ii= 44 , relres= 1.034866754951606e-10 , err= 3.010276626245948e-11
ii= 45 , relres= 5.397347825671701e-11 , err= 1.9475271121599457e-11
FULL: psf5
ii= 0 , relres= 0.4013422842141698 , err= 0.295308520820134
ii= 1 , relres= 0.18887243974645876 , err= 0.08714860047837354
ii= 2 , relres= 0.12499897401905347 , err= 0.04541191487129473
ii= 3 , relres= 0.06946294100576228 , err= 0.030539515925864073
ii= 4 , relres= 0.03932438598861431 , err= 0.015154486363764878
ii= 5 , relres= 0.0274521473466701 , err= 0.008652082345489549
ii= 6 , relres= 0.016028307811182616 , err= 0.005546462937416772
ii= 7 , relres= 0.009832422012395248 , err= 0.0036779718374021285
ii= 8 , relres= 0.0060033117156086035 , err= 0.002166259503623039
ii= 9 , relres= 0.0035526534772451685 , err= 0.0012832964348805922
ii= 10 , relres= 0.002003660724848768 , err= 0.0006886957918504659
ii= 11 , relres= 0.001191933467943935 , err= 0.0003658291287959482
ii= 12 , relres= 0.0006769733978944217 , err= 0.0002317368256797067
ii= 13 , relres= 0.000363226901734677 , err= 0.00013040012377591784

```

```

ii= 14 , relres= 0.00020089333811137592 , err= 7.84388694448379e-05
ii= 15 , relres= 0.00010983635564220537 , err= 4.288992483757609e-05
ii= 16 , relres= 6.151902957572947e-05 , err= 2.2087490935427057e-05
ii= 17 , relres= 3.565686138559539e-05 , err= 1.2298640525151175e-05
ii= 18 , relres= 2.018021789157247e-05 , err= 7.315573035493641e-06
ii= 19 , relres= 1.2200404053539589e-05 , err= 3.938268547872204e-06
ii= 20 , relres= 6.663726068412809e-06 , err= 2.332122104823159e-06
ii= 21 , relres= 3.543274124144071e-06 , err= 1.2438842619169913e-06
ii= 22 , relres= 1.717428730633117e-06 , err= 6.364574452897861e-07
ii= 23 , relres= 9.358023015722979e-07 , err= 3.132696240724536e-07
ii= 24 , relres= 4.950592771352807e-07 , err= 1.6228254320949267e-07
ii= 25 , relres= 3.0547111817963703e-07 , err= 8.914448304150637e-08
ii= 26 , relres= 1.9432931779400792e-07 , err= 5.562488636882918e-08
ii= 27 , relres= 1.0352165710872793e-07 , err= 3.541081858840282e-08
ii= 28 , relres= 6.151552622770189e-08 , err= 1.9520162035086253e-08
ii= 29 , relres= 3.6325881367433015e-08 , err= 1.1228566987933908e-08
ii= 30 , relres= 2.1285596698066583e-08 , err= 6.593512999518424e-09
ii= 31 , relres= 1.0664981923328277e-08 , err= 3.79077812160842e-09
ii= 32 , relres= 5.7734174138077964e-09 , err= 2.0067804876354983e-09
ii= 33 , relres= 2.985880613092808e-09 , err= 9.982973695283694e-10
ii= 34 , relres= 1.5544977567313397e-09 , err= 5.35152701217787e-10
ii= 35 , relres= 9.252204638964321e-10 , err= 2.79387222437591e-10
ii= 36 , relres= 5.709590881244258e-10 , err= 1.7230765772165693e-10
ii= 37 , relres= 2.959920057122784e-10 , err= 9.272629288012314e-11
ii= 38 , relres= 1.4406651167013214e-10 , err= 4.6544747962818725e-11
ii= 39 , relres= 9.940379263078667e-11 , err= 2.371436128069151e-11
FULL: psf25
ii= 0 , relres= 0.319674396935505 , err= 0.2948834506318247
ii= 1 , relres= 0.12338139967969793 , err= 0.08074918992978672
ii= 2 , relres= 0.05948674177893956 , err= 0.033287462352831886
ii= 3 , relres= 0.02873594496854482 , err= 0.016059765070489652
ii= 4 , relres= 0.01418554528833904 , err= 0.006887847337473451
ii= 5 , relres= 0.00775870120358675 , err= 0.003272430840982573
ii= 6 , relres= 0.0044343581089372665 , err= 0.0015979067592526528
ii= 7 , relres= 0.00248020041969897 , err= 0.0008041097508983786
ii= 8 , relres= 0.001324608253049718 , err= 0.00048554216630026045
ii= 9 , relres= 0.000617542597586057 , err= 0.0002483570173352983
ii= 10 , relres= 0.0003216338959501857 , err= 0.00011899374427701071
ii= 11 , relres= 0.0001549030005628696 , err= 6.347750213768601e-05
ii= 12 , relres= 7.330714805743457e-05 , err= 2.6924316338083527e-05
ii= 13 , relres= 3.583398698171643e-05 , err= 1.2944321775495414e-05
ii= 14 , relres= 1.5106764496808158e-05 , err= 5.890725216893646e-06
ii= 15 , relres= 7.500704649516748e-06 , err= 2.825625470048078e-06
ii= 16 , relres= 3.656191923008437e-06 , err= 1.4817481954025843e-06
ii= 17 , relres= 1.650325380189637e-06 , err= 7.500738171854907e-07
ii= 18 , relres= 7.623676552210472e-07 , err= 3.47444746106999e-07
ii= 19 , relres= 3.245544044215603e-07 , err= 1.51927690062081e-07
ii= 20 , relres= 1.4506954553005417e-07 , err= 6.98543470142341e-08
ii= 21 , relres= 6.083777063152004e-08 , err= 2.9453903903654925e-08
ii= 22 , relres= 2.4245697067888834e-08 , err= 1.1666953486418501e-08
ii= 23 , relres= 1.0608085407559839e-08 , err= 4.876752442384817e-09
ii= 24 , relres= 4.478103933021794e-09 , err= 1.945672153782437e-09
ii= 25 , relres= 1.7783985867929981e-09 , err= 7.231079887693173e-10
ii= 26 , relres= 7.064984678714521e-10 , err= 2.9597609764087473e-10
ii= 27 , relres= 2.7483752261740746e-10 , err= 1.239866736830855e-10
ii= 28 , relres= 1.553410117278497e-10 , err= 5.891184317166503e-11
ii= 29 , relres= 1.217151932143965e-10 , err= 2.3314906582189402e-11
ii= 30 , relres= 8.943094301368401e-11 , err= 0.0
FULL: reg
ii= 0 , relres= 1.076796185186652 , err= 0.9957643366550978
ii= 1 , relres= 1.0628435462225905 , err= 0.9907826837074074

```

```
ii= 2 , relres= 1.113866839880944 , err= 0.9820894888026643
ii= 3 , relres= 1.1741146093894266 , err= 0.968010065460746
ii= 4 , relres= 1.160270004054477 , err= 0.9560320296541491
ii= 5 , relres= 1.1720795442564922 , err= 0.9411529978724663
ii= 6 , relres= 1.182507177839601 , err= 0.928566876346975
ii= 7 , relres= 1.182891082927204 , err= 0.9100024287785276
ii= 8 , relres= 1.1969738812640727 , err= 0.895092152448145
ii= 9 , relres= 1.1361127943149272 , err= 0.8737381954429815
ii= 10 , relres= 1.1834055035430349 , err= 0.8569266208229244
ii= 11 , relres= 1.2003707159638624 , err= 0.8396815042608764
ii= 12 , relres= 1.1773489939960065 , err= 0.8120599059892861
ii= 13 , relres= 1.1814821800043909 , err= 0.7875186511453443
ii= 14 , relres= 1.1489976869582321 , err= 0.7676564472659386
ii= 15 , relres= 1.128403951181268 , err= 0.744086573705896
ii= 16 , relres= 1.1412474426683432 , err= 0.7241485965304724
ii= 17 , relres= 1.1132774945898827 , err= 0.7026650744767804
ii= 18 , relres= 1.0588778638084742 , err= 0.6741339813382041
ii= 19 , relres= 1.1064190957871627 , err= 0.646844080947726
ii= 20 , relres= 1.0554517329832167 , err= 0.6235390380063818
ii= 21 , relres= 1.0706490588129298 , err= 0.6022925257776974
ii= 22 , relres= 0.986684261711358 , err= 0.5815487172445664
ii= 23 , relres= 0.9898443703311015 , err= 0.5669250024077375
ii= 24 , relres= 0.983650758693951 , err= 0.5568138012042216
ii= 25 , relres= 0.966733680984003 , err= 0.5429507831928436
ii= 26 , relres= 0.973075025953109 , err= 0.5218767150738041
ii= 27 , relres= 0.9283298119294848 , err= 0.49727119559740474
ii= 28 , relres= 0.9025646592534576 , err= 0.4804784597697183
ii= 29 , relres= 0.8758541236481818 , err= 0.4702982317622348
ii= 30 , relres= 0.8239324838759027 , err= 0.46064698902716367
ii= 31 , relres= 0.8164683112121883 , err= 0.4477824343193291
ii= 32 , relres= 0.8188457667421125 , err= 0.43880265804581453
ii= 33 , relres= 0.7831696717192989 , err= 0.4247550468168492
ii= 34 , relres= 0.8332877950201725 , err= 0.4123825446724927
ii= 35 , relres= 0.7489368882590803 , err= 0.40393164313803764
ii= 36 , relres= 0.7147238440404902 , err= 0.3915486313744133
ii= 37 , relres= 0.6840998122054557 , err= 0.38063511653182885
ii= 38 , relres= 0.6974119904667877 , err= 0.3655268441282386
ii= 39 , relres= 0.6630647934726067 , err= 0.3511085954286283
ii= 40 , relres= 0.6843380143197999 , err= 0.3368085471356562
ii= 41 , relres= 0.6814290640731738 , err= 0.32773863910068884
ii= 42 , relres= 0.650431918651408 , err= 0.3157070295936308
ii= 43 , relres= 0.638469943581816 , err= 0.3026778762248431
ii= 44 , relres= 0.6370874387690112 , err= 0.2973688926862987
ii= 45 , relres= 0.60764651912939 , err= 0.2866056647726662
ii= 46 , relres= 0.5924925510432726 , err= 0.272398511830884
ii= 47 , relres= 0.5434372569639647 , err= 0.2555872023224116
ii= 48 , relres= 0.5689324826871672 , err= 0.24873388811502134
ii= 49 , relres= 0.5206683248671151 , err= 0.23973131187535485
ii= 50 , relres= 0.533190655503546 , err= 0.22833606437141768
ii= 51 , relres= 0.5065177117724547 , err= 0.22448297136190337
ii= 52 , relres= 0.46919458598353536 , err= 0.21473206401342504
ii= 53 , relres= 0.475557314831752 , err= 0.20622879414018563
ii= 54 , relres= 0.44802448410041595 , err= 0.20231214111665058
ii= 55 , relres= 0.454180402654377 , err= 0.19412421963946594
ii= 56 , relres= 0.419962311355944 , err= 0.19061590069868112
ii= 57 , relres= 0.401152396101159 , err= 0.1822524237209738
ii= 58 , relres= 0.3965337075953906 , err= 0.17820944479566317
ii= 59 , relres= 0.3812901115896125 , err= 0.1706149727855148
ii= 60 , relres= 0.36784942672276727 , err= 0.16273488616406453
ii= 61 , relres= 0.3564552198058591 , err= 0.15623244913860573
ii= 62 , relres= 0.34097503525550643 , err= 0.15251960144216395
```

```
ii= 63 , relres= 0.3489565747245024 , err= 0.1504409153967909
ii= 64 , relres= 0.3162689146617519 , err= 0.14392431448590928
ii= 65 , relres= 0.3094628352800348 , err= 0.13674440335877644
ii= 66 , relres= 0.3010445878536464 , err= 0.13045302497764394
ii= 67 , relres= 0.2857274282157357 , err= 0.12759172380127096
ii= 68 , relres= 0.27880551698881245 , err= 0.1220666941825721
ii= 69 , relres= 0.26318007840916935 , err= 0.1199028326791266
ii= 70 , relres= 0.25127003038681456 , err= 0.11495775017370732
ii= 71 , relres= 0.2661483502688088 , err= 0.11229614049144689
ii= 72 , relres= 0.2632772452476006 , err= 0.10910634410955207
ii= 73 , relres= 0.22951348486453632 , err= 0.10555468405688131
ii= 74 , relres= 0.22525080920717336 , err= 0.10131662517714456
ii= 75 , relres= 0.2329083812772699 , err= 0.09910661407275347
ii= 76 , relres= 0.20792754552301335 , err= 0.09587541550349847
ii= 77 , relres= 0.2156398808544991 , err= 0.0945162255816567
ii= 78 , relres= 0.2133705958055988 , err= 0.09030660025385594
ii= 79 , relres= 0.20041758791189926 , err= 0.08751235675975853
ii= 80 , relres= 0.1849502944208758 , err= 0.08330621135322762
ii= 81 , relres= 0.19546171540915883 , err= 0.07939589206854647
ii= 82 , relres= 0.18978687024302793 , err= 0.07737124349776248
ii= 83 , relres= 0.17835550621903673 , err= 0.07600124118020762
ii= 84 , relres= 0.17513762838554367 , err= 0.07334848375397728
ii= 85 , relres= 0.16841627547153104 , err= 0.07148036747028055
ii= 86 , relres= 0.1644381171308193 , err= 0.06801077720941501
ii= 87 , relres= 0.15437422476137316 , err= 0.0661567276086849
ii= 88 , relres= 0.15095296294607033 , err= 0.06269220825623355
ii= 89 , relres= 0.1428666347367706 , err= 0.05988205085971108
ii= 90 , relres= 0.14608076279621038 , err= 0.057719204124368644
ii= 91 , relres= 0.13162872926309346 , err= 0.05484833496194037
ii= 92 , relres= 0.12997261774562005 , err= 0.05427014359788193
ii= 93 , relres= 0.12340007170811258 , err= 0.05173443459369165
ii= 94 , relres= 0.12016298533918539 , err= 0.05059651500379105
ii= 95 , relres= 0.12285098781205532 , err= 0.04875630494021213
ii= 96 , relres= 0.11174665540098841 , err= 0.04770372883029235
ii= 97 , relres= 0.11655412019878567 , err= 0.04670855886410051
ii= 98 , relres= 0.11360028435143124 , err= 0.04490563880453454
ii= 99 , relres= 0.10807598577728458 , err= 0.043918306339544425
ii= 100 , relres= 0.10396526021478367 , err= 0.042992896781648596
ii= 101 , relres= 0.09671437542120942 , err= 0.04085447401492392
ii= 102 , relres= 0.09759267886940023 , err= 0.04036883790851838
ii= 103 , relres= 0.09774861178263017 , err= 0.03876427336766029
ii= 104 , relres= 0.0875325759198125 , err= 0.03752665213559114
ii= 105 , relres= 0.08665048852150802 , err= 0.03564258284571925
ii= 106 , relres= 0.08100359002739112 , err= 0.03467967310444956
ii= 107 , relres= 0.08016742331004763 , err= 0.033077935320576145
ii= 108 , relres= 0.07882628834170521 , err= 0.03244034925062141
ii= 109 , relres= 0.07346686457599312 , err= 0.03080549384755617
ii= 110 , relres= 0.07468400017608816 , err= 0.02955616954853785
ii= 111 , relres= 0.07072236399897758 , err= 0.028319471373360067
ii= 112 , relres= 0.07105939680395659 , err= 0.02762094768012525
ii= 113 , relres= 0.0696956308536372 , err= 0.02655348066158965
ii= 114 , relres= 0.06295041756830717 , err= 0.02612725136839595
ii= 115 , relres= 0.06292459940746022 , err= 0.02478982694351635
ii= 116 , relres= 0.06005463556422758 , err= 0.024195896384445564
ii= 117 , relres= 0.05834392542234621 , err= 0.023770586523150245
ii= 118 , relres= 0.06160346903763441 , err= 0.02333767078596238
ii= 119 , relres= 0.05356584357338975 , err= 0.02224393529731127
ii= 120 , relres= 0.0571895654379659 , err= 0.02165300654836154
ii= 121 , relres= 0.05115721632429108 , err= 0.020783917654175983
ii= 122 , relres= 0.05566098430565886 , err= 0.020024439505084125
ii= 123 , relres= 0.053106971084391545 , err= 0.0195064571766019
```

```
ii= 124 , relres= 0.04787108614246064 , err= 0.0188675730547446
ii= 125 , relres= 0.04787916980853606 , err= 0.018406123213119553
ii= 126 , relres= 0.04572758530058755 , err= 0.017648768684217147
ii= 127 , relres= 0.04604984186022236 , err= 0.017359980911545378
ii= 128 , relres= 0.042150165274874335 , err= 0.016318923256346277
ii= 129 , relres= 0.041172031447196604 , err= 0.01578718602215573
ii= 130 , relres= 0.03721714125698223 , err= 0.014922099124148862
ii= 131 , relres= 0.03613184424413992 , err= 0.014201350151905453
ii= 132 , relres= 0.03342221992089164 , err= 0.013880938391842729
ii= 133 , relres= 0.03523811555007934 , err= 0.013576163842944832
ii= 134 , relres= 0.034807796302523474 , err= 0.013310843601417061
ii= 135 , relres= 0.033479733806887575 , err= 0.012757271232318911
ii= 136 , relres= 0.033048797752218674 , err= 0.012624446622334843
ii= 137 , relres= 0.03120115915969755 , err= 0.011784876263504974
ii= 138 , relres= 0.029007618735487656 , err= 0.011579100849995205
ii= 139 , relres= 0.02944604623066612 , err= 0.011020129972954361
ii= 140 , relres= 0.028745201877750284 , err= 0.010730162352667966
ii= 141 , relres= 0.02681521896481728 , err= 0.010482195804535663
ii= 142 , relres= 0.02774207515547366 , err= 0.010010086810649204
ii= 143 , relres= 0.02609341335578342 , err= 0.009819382873795275
ii= 144 , relres= 0.025104890070742434 , err= 0.009668863474555093
ii= 145 , relres= 0.024279290905480602 , err= 0.009010664195229915
ii= 146 , relres= 0.023788481032395115 , err= 0.008815146830327412
ii= 147 , relres= 0.024189183984287363 , err= 0.00841104432966562
ii= 148 , relres= 0.0215721231620789 , err= 0.00810697732650997
ii= 149 , relres= 0.0207470812657205 , err= 0.00784064486883577
ii= 150 , relres= 0.02041133295069573 , err= 0.007608745171112141
ii= 151 , relres= 0.018614176913665532 , err= 0.00735658734659848
ii= 152 , relres= 0.020399114741202987 , err= 0.007166975892017684
ii= 153 , relres= 0.017867942396908337 , err= 0.006847007643792122
ii= 154 , relres= 0.017510475923839905 , err= 0.0067820460740389815
ii= 155 , relres= 0.01750390455289768 , err= 0.0065814561531443725
ii= 156 , relres= 0.01676805625587702 , err= 0.006358691097756212
ii= 157 , relres= 0.016858598843117738 , err= 0.00616926702832702
ii= 158 , relres= 0.016079746196187503 , err= 0.00609122568203382
ii= 159 , relres= 0.015053803320423529 , err= 0.005731615106257067
ii= 160 , relres= 0.01476286597853722 , err= 0.005613352241145947
ii= 161 , relres= 0.014477612636685082 , err= 0.005393682730314205
ii= 162 , relres= 0.01373059903179006 , err= 0.005260032073491905
ii= 163 , relres= 0.013957894379413752 , err= 0.0050386862831223725
ii= 164 , relres= 0.012742775498976643 , err= 0.004941884149756683
ii= 165 , relres= 0.012779270791890007 , err= 0.004822038778656623
ii= 166 , relres= 0.011944860283778861 , err= 0.004748155428383373
ii= 167 , relres= 0.01147180089590037 , err= 0.004521434617858802
ii= 168 , relres= 0.010693833218490444 , err= 0.004413448423439598
ii= 169 , relres= 0.010489755227268225 , err= 0.0041439429155937235
ii= 170 , relres= 0.010190493162398413 , err= 0.004025594324456015
ii= 171 , relres= 0.010685823956599737 , err= 0.003901436486795894
ii= 172 , relres= 0.01018760781240041 , err= 0.003827284376567981
ii= 173 , relres= 0.009141992757650019 , err= 0.003699427965895211
ii= 174 , relres= 0.009511409472104963 , err= 0.003616559736816939
ii= 175 , relres= 0.00864644732609575 , err= 0.003435636732985137
ii= 176 , relres= 0.009191587060169752 , err= 0.00331869380363874
ii= 177 , relres= 0.008875821348656932 , err= 0.0032509547757226635
ii= 178 , relres= 0.009011247792726784 , err= 0.003119391453868915
ii= 179 , relres= 0.007921796385254537 , err= 0.003028631277600484
ii= 180 , relres= 0.007803909479641662 , err= 0.002869139777187672
ii= 181 , relres= 0.007157923895134037 , err= 0.0028169493914328415
ii= 182 , relres= 0.007107064307840152 , err= 0.0027190595337611393
ii= 183 , relres= 0.006682231287842167 , err= 0.0026132682013628294
ii= 184 , relres= 0.0065830481845301795 , err= 0.0025795125629675913
```

```
ii= 185 , relres= 0.006487941454486128 , err= 0.002516248634761274
ii= 186 , relres= 0.006886423528637071 , err= 0.0024316960440745107
ii= 187 , relres= 0.006120884405165249 , err= 0.002332527957404464
ii= 188 , relres= 0.006182321900898902 , err= 0.0022768079173392268
ii= 189 , relres= 0.005727277036930206 , err= 0.0021743318817038122
ii= 190 , relres= 0.005696387680737971 , err= 0.0020899666103518727
ii= 191 , relres= 0.005412966148594127 , err= 0.002061959925810266
ii= 192 , relres= 0.005538056201008128 , err= 0.002017070987738002
ii= 193 , relres= 0.00511395683881196 , err= 0.0019224857025984586
ii= 194 , relres= 0.005066150926159309 , err= 0.0018876812360922974
ii= 195 , relres= 0.00463348953589368 , err= 0.0017947106700018638
ii= 196 , relres= 0.004844560738978691 , err= 0.0017308246367119052
ii= 197 , relres= 0.004394915450787895 , err= 0.0016383626543516262
ii= 198 , relres= 0.004452873967271022 , err= 0.0016165972318352001
ii= 199 , relres= 0.004510319070066649 , err= 0.0015803262180562087
ii= 200 , relres= 0.004052538618698687 , err= 0.0015109614865717606
ii= 201 , relres= 0.003964736825123303 , err= 0.0014688279920382643
ii= 202 , relres= 0.0037909999963200627 , err= 0.001432953651613372
ii= 203 , relres= 0.0036138969299395717 , err= 0.0013787134710895282
ii= 204 , relres= 0.003498916429625585 , err= 0.0013362294190949052
ii= 205 , relres= 0.0036233520662190253 , err= 0.0013178548242000626
ii= 206 , relres= 0.0036931597544305242 , err= 0.0012685610511965276
ii= 207 , relres= 0.0033075573995735483 , err= 0.001226577211841309
ii= 208 , relres= 0.0030988472813870903 , err= 0.0012036510144411983
ii= 209 , relres= 0.0032909438766027224 , err= 0.0011823375138721348
ii= 210 , relres= 0.0030734163803345176 , err= 0.0011443566255509943
ii= 211 , relres= 0.00308935717152172 , err= 0.0011247581423282668
ii= 212 , relres= 0.002658336949271635 , err= 0.001083668843948576
ii= 213 , relres= 0.002725284366900626 , err= 0.0010260069733995308
ii= 214 , relres= 0.0026805254289846724 , err= 0.0010007123699276286
ii= 215 , relres= 0.0024705047843298004 , err= 0.0009711672023356315
ii= 216 , relres= 0.0025605350123566427 , err= 0.0009370805407063515
ii= 217 , relres= 0.0023051511295896026 , err= 0.0008966920538257297
ii= 218 , relres= 0.0023192638784715843 , err= 0.0008604949525337356
ii= 219 , relres= 0.0021623866721855105 , err= 0.000817779246130896
ii= 220 , relres= 0.0021287999218662867 , err= 0.000807784911714495
ii= 221 , relres= 0.002008436035857144 , err= 0.0007819862918813343
ii= 222 , relres= 0.0018795788905993443 , err= 0.0007220358225133842
ii= 223 , relres= 0.0019146974325444633 , err= 0.0007073909389069479
ii= 224 , relres= 0.001887733061431669 , err= 0.0006742121969827536
ii= 225 , relres= 0.0017417223810138956 , err= 0.00066355724124205
ii= 226 , relres= 0.0016834177550390655 , err= 0.0006426791395399144
ii= 227 , relres= 0.0017919764162613035 , err= 0.0006157617050131315
ii= 228 , relres= 0.0018338649163408645 , err= 0.0005963194900442551
ii= 229 , relres= 0.0016400510005463043 , err= 0.000569008804421656
ii= 230 , relres= 0.0015614313589352722 , err= 0.0005570026477700247
ii= 231 , relres= 0.0014772705937195687 , err= 0.0005488052351920538
ii= 232 , relres= 0.0015587503215449747 , err= 0.0005299907408279976
ii= 233 , relres= 0.0014321072168894458 , err= 0.0005064092681261775
ii= 234 , relres= 0.0014278299918217333 , err= 0.0004979680678300171
ii= 235 , relres= 0.0014026345372391788 , err= 0.0004744202698879181
ii= 236 , relres= 0.001364999851893021 , err= 0.0004558158679463855
ii= 237 , relres= 0.0012793959331606965 , err= 0.00044106381975280546
ii= 238 , relres= 0.0012080840505082732 , err= 0.0004308171593910245
ii= 239 , relres= 0.0011859666388060594 , err= 0.0003989838021817248
ii= 240 , relres= 0.0010716761775879113 , err= 0.00038763618755103597
ii= 241 , relres= 0.0012230796470788566 , err= 0.0003742854333768273
ii= 242 , relres= 0.0010900349675708166 , err= 0.0003580324030354213
ii= 243 , relres= 0.0009794075107767918 , err= 0.00035181426386868053
ii= 244 , relres= 0.0010120213446347703 , err= 0.0003449774885209867
ii= 245 , relres= 0.0010915608081719709 , err= 0.000334719992984649
```

```

ii= 246 , relres= 0.0009029516476656963 , err= 0.00032030653473866553
ii= 247 , relres= 0.0008801736214546742 , err= 0.0003158565593636253
ii= 248 , relres= 0.0008536589195471538 , err= 0.00030281929364551975
ii= 249 , relres= 0.0008105639101210396 , err= 0.0002872223716328129
ii= 250 , relres= 0.0007728294793320696 , err= 0.0002816688829917392
ii= 251 , relres= 0.0008609995343715391 , err= 0.0002706489937188563
ii= 252 , relres= 0.0007483983271237035 , err= 0.0002630378433472067
ii= 253 , relres= 0.0006965845835081785 , err= 0.00025650133301067564
ii= 254 , relres= 0.0006675527506995766 , err= 0.00025354137699997603
ii= 255 , relres= 0.000682647918482602 , err= 0.00024476507167689796
ii= 256 , relres= 0.0006718778298187567 , err= 0.00023724778037827795
ii= 257 , relres= 0.0005920845231239347 , err= 0.00022781598991583257
ii= 258 , relres= 0.0006404150979475193 , err= 0.00021766989652791027
ii= 259 , relres= 0.0006073084681818645 , err= 0.00020860351647372392
ii= 260 , relres= 0.00054706984389624 , err= 0.00020447178085383338
ii= 261 , relres= 0.0005328438585931197 , err= 0.00019639923514316388
ii= 262 , relres= 0.0005674869267717713 , err= 0.00019357311396409798
ii= 263 , relres= 0.0005407308092442178 , err= 0.00018309674157656274
ii= 264 , relres= 0.0004871166604738923 , err= 0.0001779671343881593
ii= 265 , relres= 0.0005167705637208793 , err= 0.00016886635768255606
ii= 266 , relres= 0.0004560132140849334 , err= 0.00016544793529244968
ii= 267 , relres= 0.0004310364228926071 , err= 0.00015945163519449213
ii= 268 , relres= 0.00041797961136670673 , err= 0.0001514219015850322
ii= 269 , relres= 0.00040419998646564127 , err= 0.00014457369951249093
ii= 270 , relres= 0.0003796968450929143 , err= 0.0001420137815642647
ii= 271 , relres= 0.00041157297571845755 , err= 0.00013612077159355906
ii= 272 , relres= 0.0003764542270403787 , err= 0.00013230251759839066
ii= 273 , relres= 0.0003578752880687821 , err= 0.00012998451255183575
ii= 274 , relres= 0.0003580106983457154 , err= 0.00012607228396347363
ii= 275 , relres= 0.00035186852344478575 , err= 0.00012227949720394787
ii= 276 , relres= 0.00032741295366588445 , err= 0.00011938428500075274
ii= 277 , relres= 0.00032105193510429357 , err= 0.00011730603972596334
ii= 278 , relres= 0.0003272768847095764 , err= 0.00011247273823248068
ii= 279 , relres= 0.00031794130821085904 , err= 0.00010750236905576014
ii= 280 , relres= 0.0003023096336463378 , err= 0.00010383531748652551
ii= 281 , relres= 0.0002961245112902743 , err= 0.00010092371954679081
ii= 282 , relres= 0.0002814520124866321 , err= 9.943494881755775e-05
ii= 283 , relres= 0.00029379165710569056 , err= 9.632028193298672e-05
ii= 284 , relres= 0.0002487372140933521 , err= 9.08058803319404e-05
ii= 285 , relres= 0.0002573318019407451 , err= 8.846796110262722e-05
ii= 286 , relres= 0.0002677799740421897 , err= 8.543210505441267e-05
ii= 287 , relres= 0.00024327230120008573 , err= 8.110773220947809e-05
ii= 288 , relres= 0.0002350573484904168 , err= 8.014240688140511e-05
ii= 289 , relres= 0.00023300328730450635 , err= 7.694536133962061e-05
ii= 290 , relres= 0.00022270060202950296 , err= 7.479876032042439e-05
ii= 291 , relres= 0.00020944175815906376 , err= 7.154221045745433e-05
ii= 292 , relres= 0.00019536678719827384 , err= 6.818962058352464e-05
ii= 293 , relres= 0.00018017755018291895 , err= 6.359988644168634e-05
ii= 294 , relres= 0.00017055525059681595 , err= 6.312705383672426e-05
ii= 295 , relres= 0.00017668983161784493 , err= 6.108734630677725e-05
ii= 296 , relres= 0.00016325814236309627 , err= 5.869852359851578e-05
ii= 297 , relres= 0.0001523735554946196 , err= 5.668114298505774e-05
ii= 298 , relres= 0.00015094782660368095 , err= 5.603638721714037e-05
ii= 299 , relres= 0.00015078528379115997 , err= 5.458365040574074e-05
ii= 300 , relres= 0.00014920780867351912 , err= 5.3852624225241794e-05
ii= 301 , relres= 0.0001439763396329965 , err= 5.232517527736222e-05
ii= 302 , relres= 0.00013523833424535119 , err= 5.094456444514937e-05
ii= 303 , relres= 0.00014075557046579372 , err= 4.9991856239383604e-05
ii= 304 , relres= 0.00014359110796251175 , err= 4.714663474093993e-05
ii= 305 , relres= 0.00012157762499715191 , err= 4.573616943001965e-05
ii= 306 , relres= 0.00012683238509127157 , err= 4.429940482264784e-05

```

```

ii= 307 , relres= 0.00012514634111090802 , err= 4.123065477045732e-05
ii= 308 , relres= 0.00011576372988064573 , err= 4.016638507864904e-05
ii= 309 , relres= 0.00011543621616502627 , err= 3.96246748466484e-05
ii= 310 , relres= 0.00010860740653494491 , err= 3.81601316092777e-05
ii= 311 , relres= 0.00011083298453603967 , err= 3.7324714043974966e-05
ii= 312 , relres= 0.00010432034768957569 , err= 3.555937260531468e-05
ii= 313 , relres= 9.576233217022554e-05 , err= 3.47026648671868e-05
ii= 314 , relres= 9.796912198284254e-05 , err= 3.263422334601208e-05
ii= 315 , relres= 9.515951291463125e-05 , err= 3.222896939524438e-05
ii= 316 , relres= 8.707958588419649e-05 , err= 3.067484605019088e-05
ii= 317 , relres= 8.443234794400078e-05 , err= 2.9944388990987438e-05
ii= 318 , relres= 8.53703944686951e-05 , err= 2.9495214812074735e-05
ii= 319 , relres= 8.520299665112035e-05 , err= 2.8645570157717176e-05
ii= 320 , relres= 8.003139627990064e-05 , err= 2.735354233707287e-05
ii= 321 , relres= 7.29384937078094e-05 , err= 2.663606526428945e-05
ii= 322 , relres= 7.18943145870836e-05 , err= 2.596597293941758e-05
ii= 323 , relres= 7.537455154403386e-05 , err= 2.5263747889912898e-05
ii= 324 , relres= 6.556997551178483e-05 , err= 2.4326588612296034e-05
ii= 325 , relres= 6.549211496179838e-05 , err= 2.398295733058284e-05
ii= 326 , relres= 6.509563995116246e-05 , err= 2.254760648923971e-05
ii= 327 , relres= 5.827013286758466e-05 , err= 2.159693079827716e-05
ii= 328 , relres= 5.8136547898316506e-05 , err= 2.131910311938381e-05
ii= 329 , relres= 5.6634643375698054e-05 , err= 2.0237257878999225e-05
ii= 330 , relres= 5.7318901291319066e-05 , err= 1.9923233389606542e-05
ii= 331 , relres= 5.759816097868536e-05 , err= 1.9505766514014032e-05
ii= 332 , relres= 5.4105683583448044e-05 , err= 1.8869555318789262e-05
ii= 333 , relres= 5.1679706546346757e-05 , err= 1.8252078298717486e-05
ii= 334 , relres= 5.324384610200385e-05 , err= 1.7322339552376072e-05
ii= 335 , relres= 4.905735252981087e-05 , err= 1.700557080797816e-05
ii= 336 , relres= 4.686423728080285e-05 , err= 1.672332826846754e-05
ii= 337 , relres= 4.65265698879015e-05 , err= 1.640414999823879e-05
ii= 338 , relres= 4.946195062936692e-05 , err= 1.574560036917337e-05
ii= 339 , relres= 4.786596584853093e-05 , err= 1.530834351707789e-05
ii= 340 , relres= 4.1558412711547186e-05 , err= 1.4726682433434967e-05
ii= 341 , relres= 3.916702030675721e-05 , err= 1.452133058972072e-05
ii= 342 , relres= 4.094923816995063e-05 , err= 1.3786847565690922e-05
ii= 343 , relres= 3.88188679841205e-05 , err= 1.3356562198527377e-05
ii= 344 , relres= 3.598088331211483e-05 , err= 1.2529779503720095e-05
ii= 345 , relres= 3.754672361062605e-05 , err= 1.1769645866047727e-05
ii= 346 , relres= 3.4563984388313153e-05 , err= 1.1514906276885601e-05
ii= 347 , relres= 3.1239159147904785e-05 , err= 1.1104909451700823e-05
ii= 348 , relres= 3.0760954905065654e-05 , err= 1.0583646056808303e-05
ii= 349 , relres= 2.9104858185547576e-05 , err= 1.020711242536209e-05
ii= 350 , relres= 2.88773582235942e-05 , err= 9.799529098984617e-06
ii= 351 , relres= 2.9610897406448614e-05 , err= 9.57661356676692e-06
ii= 352 , relres= 2.8181625180265033e-05 , err= 9.357599068509093e-06
ii= 353 , relres= 2.842953061730231e-05 , err= 8.988312619047783e-06
ii= 354 , relres= 2.647568662159785e-05 , err= 8.669544194458664e-06
ii= 355 , relres= 2.5502817078397157e-05 , err= 8.540394625121856e-06
ii= 356 , relres= 2.4319929886349218e-05 , err= 8.248412924772711e-06
ii= 357 , relres= 2.4828744220474062e-05 , err= 7.755470989528052e-06
ii= 358 , relres= 2.2901511358035203e-05 , err= 7.650913560842841e-06
ii= 359 , relres= 2.2660772729041917e-05 , err= 7.437583978292758e-06
ii= 360 , relres= 2.0819836424176948e-05 , err= 7.238537777664032e-06
ii= 361 , relres= 2.2607387508333702e-05 , err= 7.065667724402164e-06
ii= 362 , relres= 2.0405390955763638e-05 , err= 6.696619886824949e-06
ii= 363 , relres= 1.826967342923351e-05 , err= 6.571405021926233e-06
ii= 364 , relres= 1.8764400587078386e-05 , err= 6.3857428716408955e-06
ii= 365 , relres= 1.8614135915990138e-05 , err= 6.215543103076465e-06
ii= 366 , relres= 1.8454193763403627e-05 , err= 5.966647906911305e-06
ii= 367 , relres= 1.641001237337024e-05 , err= 5.7283062735841496e-06

```

```

ii= 368 , relres= 1.686178978445669e-05 , err= 5.598074985362156e-06
ii= 369 , relres= 1.696016927922357e-05 , err= 5.352881517946835e-06
ii= 370 , relres= 1.6275207773942144e-05 , err= 5.211809376731723e-06
ii= 371 , relres= 1.5365723268991573e-05 , err= 5.097139336601053e-06
ii= 372 , relres= 1.4940325481779992e-05 , err= 4.8979268020737806e-06
ii= 373 , relres= 1.371558030201141e-05 , err= 4.841559950221259e-06
ii= 374 , relres= 1.380008211179392e-05 , err= 4.763291634035542e-06
ii= 375 , relres= 1.3358848706849656e-05 , err= 4.538257413100434e-06
ii= 376 , relres= 1.1487917752000915e-05 , err= 4.429273909863195e-06
ii= 377 , relres= 1.2189931702316951e-05 , err= 4.289557642339898e-06
ii= 378 , relres= 1.2270482737386283e-05 , err= 4.168061810431519e-06
ii= 379 , relres= 1.132335959894569e-05 , err= 4.085683087725097e-06
ii= 380 , relres= 1.061077313074552e-05 , err= 3.955929413787619e-06
ii= 381 , relres= 1.1085126241125902e-05 , err= 3.870566094059903e-06
ii= 382 , relres= 9.987801305989347e-06 , err= 3.8059244465538704e-06
ii= 383 , relres= 9.820937247251081e-06 , err= 3.5777486566847414e-06
ii= 384 , relres= 9.329957351579263e-06 , err= 3.5117285265116093e-06
ii= 385 , relres= 9.522831147083154e-06 , err= 3.395081333817498e-06
ii= 386 , relres= 9.95133286656743e-06 , err= 3.244553545810572e-06
ii= 387 , relres= 8.52213006084563e-06 , err= 3.0552086968053824e-06
ii= 388 , relres= 8.388125089170196e-06 , err= 3.0191811909721795e-06
ii= 389 , relres= 8.222900882654261e-06 , err= 2.90445648242709e-06
ii= 390 , relres= 8.099093350629769e-06 , err= 2.7962034124961158e-06
ii= 391 , relres= 7.869276135760474e-06 , err= 2.7612082869992567e-06
ii= 392 , relres= 7.144783676754632e-06 , err= 2.6290018262575177e-06
ii= 393 , relres= 7.561209277568827e-06 , err= 2.555599064293234e-06
ii= 394 , relres= 7.328956031404301e-06 , err= 2.4450128245313194e-06
ii= 395 , relres= 6.670971983744453e-06 , err= 2.3644669695804695e-06
ii= 396 , relres= 6.38404477955298e-06 , err= 2.318583631027447e-06
ii= 397 , relres= 6.429974706423402e-06 , err= 2.2597078229663214e-06
ii= 398 , relres= 6.046070970938423e-06 , err= 2.2136456696441946e-06
ii= 399 , relres= 7.351915589283336e-06 , err= 2.1133945551059645e-06
ii= 400 , relres= 5.756580532205795e-06 , err= 2.007764274594122e-06
ii= 401 , relres= 5.460525210015097e-06 , err= 1.9452125441763745e-06
ii= 402 , relres= 5.519540226130115e-06 , err= 1.928446953960286e-06
ii= 403 , relres= 5.589301033206924e-06 , err= 1.8381860096974018e-06
ii= 404 , relres= 5.576255796194124e-06 , err= 1.7646592122142037e-06
ii= 405 , relres= 4.9639561820552044e-06 , err= 1.7023637501382866e-06
ii= 406 , relres= 4.895599880246784e-06 , err= 1.6514244515629498e-06
ii= 407 , relres= 4.4862783372055734e-06 , err= 1.590832036428506e-06
ii= 408 , relres= 4.799895240772324e-06 , err= 1.5264692344638345e-06
ii= 409 , relres= 4.565422168180625e-06 , err= 1.4907870403817512e-06
ii= 410 , relres= 4.204655590067132e-06 , err= 1.4611266217478473e-06
ii= 411 , relres= 4.138591597777794e-06 , err= 1.416025954112583e-06
ii= 412 , relres= 4.185334252901848e-06 , err= 1.3654659984957752e-06
ii= 413 , relres= 3.823451172459028e-06 , err= 1.3198764814048343e-06
ii= 414 , relres= 3.6656514242105274e-06 , err= 1.2907330711773037e-06
ii= 415 , relres= 3.41833182194022e-06 , err= 1.2718986447658283e-06
ii= 416 , relres= 3.267094291802801e-06 , err= 1.1830042155176965e-06
ii= 417 , relres= 3.041377750014759e-06 , err= 1.1635238140115998e-06
ii= 418 , relres= 3.168530619070162e-06 , err= 1.1145556804293136e-06
ii= 419 , relres= 2.929697249315117e-06 , err= 1.1012182735513226e-06
ii= 420 , relres= 3.2133516380975385e-06 , err= 1.0587285433593997e-06
ii= 421 , relres= 2.934080293633544e-06 , err= 1.0215844115147362e-06
ii= 422 , relres= 2.6223792756341502e-06 , err= 1.0007515054686192e-06
ii= 423 , relres= 2.709752837576962e-06 , err= 9.65534151933479e-07
ii= 424 , relres= 2.9202766602847715e-06 , err= 9.350268223276617e-07
ii= 425 , relres= 2.431147282382979e-06 , err= 8.98896090528244e-07
ii= 426 , relres= 2.4656536343515446e-06 , err= 8.723456945516628e-07
ii= 427 , relres= 2.2842300769619697e-06 , err= 8.508617658266726e-07
ii= 428 , relres= 2.2643919395561914e-06 , err= 8.382795259605477e-07

```

```
ii= 429 , relres= 2.364407708703587e-06 , err= 8.254652887055976e-07
ii= 430 , relres= 2.3309877549115206e-06 , err= 7.88455837189103e-07
ii= 431 , relres= 1.9067360107587457e-06 , err= 7.492706534941658e-07
ii= 432 , relres= 1.9110165263865757e-06 , err= 7.248797861327853e-07
ii= 433 , relres= 1.92418324156451e-06 , err= 7.047143300614995e-07
ii= 434 , relres= 1.8409463096498803e-06 , err= 6.727097850075091e-07
ii= 435 , relres= 1.6926471950662186e-06 , err= 6.551946447036888e-07
ii= 436 , relres= 1.6388314406167766e-06 , err= 6.278976878917435e-07
ii= 437 , relres= 1.6541598805642486e-06 , err= 6.108169230824279e-07
ii= 438 , relres= 1.7287403314481867e-06 , err= 5.922693872808276e-07
ii= 439 , relres= 1.5951380436701507e-06 , err= 5.689241607997624e-07
ii= 440 , relres= 1.4766520864903036e-06 , err= 5.622432680259082e-07
ii= 441 , relres= 1.5064076927384775e-06 , err= 5.357644183484416e-07
ii= 442 , relres= 1.4365259087572302e-06 , err= 5.194961671492509e-07
ii= 443 , relres= 1.5383099796911347e-06 , err= 5.046377895907195e-07
ii= 444 , relres= 1.258646297833734e-06 , err= 4.802477084887929e-07
ii= 445 , relres= 1.277020130793378e-06 , err= 4.7370488999105567e-07
ii= 446 , relres= 1.22717125860625e-06 , err= 4.6523789912015645e-07
ii= 447 , relres= 1.2354410331657855e-06 , err= 4.5721348621554775e-07
ii= 448 , relres= 1.3338069396564676e-06 , err= 4.4069463452014026e-07
ii= 449 , relres= 1.22389875631116e-06 , err= 4.2167145087618063e-07
ii= 450 , relres= 1.0809810364502617e-06 , err= 4.03701924174004e-07
ii= 451 , relres= 1.073848712863608e-06 , err= 3.9329461249900325e-07
ii= 452 , relres= 1.0757040397018412e-06 , err= 3.768339303062147e-07
ii= 453 , relres= 1.047794085562995e-06 , err= 3.6025037699252166e-07
ii= 454 , relres= 1.0030576909487676e-06 , err= 3.485232671509656e-07
ii= 455 , relres= 9.821080964974665e-07 , err= 3.35413315510112e-07
ii= 456 , relres= 9.056243486364531e-07 , err= 3.2870295179979745e-07
ii= 457 , relres= 9.198625543346161e-07 , err= 3.1852494879481154e-07
ii= 458 , relres= 8.533652921291257e-07 , err= 2.9453521965039834e-07
ii= 459 , relres= 7.854716395289272e-07 , err= 2.893971459509501e-07
ii= 460 , relres= 8.069748732154621e-07 , err= 2.850600180768135e-07
ii= 461 , relres= 7.838319650643118e-07 , err= 2.8037142252413397e-07
ii= 462 , relres= 7.175440916578863e-07 , err= 2.569220810907374e-07
ii= 463 , relres= 7.243352202453782e-07 , err= 2.529181142860984e-07
ii= 464 , relres= 7.425929862447964e-07 , err= 2.467750736931433e-07
ii= 465 , relres= 6.668161169622247e-07 , err= 2.3768357112518629e-07
ii= 466 , relres= 7.261259183827987e-07 , err= 2.2683462825995397e-07
ii= 467 , relres= 6.661668787679859e-07 , err= 2.196599615466462e-07
ii= 468 , relres= 6.111739931222386e-07 , err= 2.0917886091880262e-07
ii= 469 , relres= 5.724450397271456e-07 , err= 2.0587697493962256e-07
ii= 470 , relres= 5.876358220569715e-07 , err= 2.029396470458098e-07
ii= 471 , relres= 5.758321176115677e-07 , err= 1.9412121092996265e-07
ii= 472 , relres= 6.016068863492919e-07 , err= 1.8983865996466572e-07
ii= 473 , relres= 4.839254721448379e-07 , err= 1.7763264768988017e-07
ii= 474 , relres= 5.101339280190448e-07 , err= 1.667568303202399e-07
ii= 475 , relres= 4.7653062415849456e-07 , err= 1.604254982035055e-07
ii= 476 , relres= 4.780996535188613e-07 , err= 1.5137137693136188e-07
ii= 477 , relres= 4.181173204785807e-07 , err= 1.4607764871339014e-07
ii= 478 , relres= 4.0744564761608227e-07 , err= 1.420521838105315e-07
ii= 479 , relres= 4.254057569301399e-07 , err= 1.3779335080700288e-07
ii= 480 , relres= 3.9041563389098793e-07 , err= 1.3478361780491472e-07
ii= 481 , relres= 3.825541958466867e-07 , err= 1.2997211595644503e-07
ii= 482 , relres= 3.6426903392658353e-07 , err= 1.2850063027471706e-07
ii= 483 , relres= 3.614329906651714e-07 , err= 1.2613615274003623e-07
ii= 484 , relres= 3.6819544925696035e-07 , err= 1.2256912062181896e-07
ii= 485 , relres= 3.5876143960562024e-07 , err= 1.1810457646597609e-07
ii= 486 , relres= 3.186060660102205e-07 , err= 1.136750712752952e-07
ii= 487 , relres= 3.233971877643532e-07 , err= 1.1035999620342992e-07
ii= 488 , relres= 3.0748289488572447e-07 , err= 1.0738303965170479e-07
ii= 489 , relres= 2.99473367163999e-07 , err= 1.040614088059497e-07
```

```

ii= 490 , relres= 3.097018913143049e-07 , err= 1.0140870015466622e-07
ii= 491 , relres= 3.086830392611309e-07 , err= 9.773972962372659e-08
ii= 492 , relres= 2.5480618767802836e-07 , err= 9.457496925816854e-08
ii= 493 , relres= 2.5875143543051617e-07 , err= 9.384921538593699e-08
ii= 494 , relres= 2.834707827966729e-07 , err= 9.063113996566338e-08
ii= 495 , relres= 2.8008894853941156e-07 , err= 8.794436965404824e-08
ii= 496 , relres= 2.2875639839414031e-07 , err= 8.125243718235575e-08
ii= 497 , relres= 2.2093076203931076e-07 , err= 7.852219999867671e-08
ii= 498 , relres= 2.2654394942297482e-07 , err= 7.500534969146592e-08
ii= 499 , relres= 2.4045681564448636e-07 , err= 7.33722969386424e-08
ii= 500 , relres= 2.0093742909442206e-07 , err= 6.999824333292532e-08
ii= 501 , relres= 1.995084249822434e-07 , err= 6.82238393739778e-08
ii= 502 , relres= 1.8630202403214837e-07 , err= 6.61567294202907e-08
ii= 503 , relres= 1.8663301450331206e-07 , err= 6.494196722256084e-08
ii= 504 , relres= 1.8331113562139635e-07 , err= 6.401264830291636e-08
ii= 505 , relres= 1.8738829785866356e-07 , err= 6.07254804059788e-08
ii= 506 , relres= 1.5901421954582827e-07 , err= 5.821837022010719e-08
ii= 507 , relres= 1.5273057655922882e-07 , err= 5.757170418998905e-08
ii= 508 , relres= 1.6503170771614197e-07 , err= 5.3982536808792386e-08
ii= 509 , relres= 1.4816472550378625e-07 , err= 5.2117608278143363e-08
ii= 510 , relres= 1.406088177535328e-07 , err= 5.0333530622090916e-08
ii= 511 , relres= 1.371999784446275e-07 , err= 4.9476983908663956e-08
ii= 512 , relres= 1.4748917550995275e-07 , err= 4.846292198695828e-08
ii= 513 , relres= 1.2967543230425466e-07 , err= 4.621134710761075e-08
ii= 514 , relres= 1.2820196544855453e-07 , err= 4.5752095514404914e-08
ii= 515 , relres= 1.236080352908889e-07 , err= 4.4215650760249364e-08
ii= 516 , relres= 1.2056501490138408e-07 , err= 4.354702741795096e-08
ii= 517 , relres= 1.1818899616997381e-07 , err= 4.214889842928301e-08
ii= 518 , relres= 1.2203472786734039e-07 , err= 4.139223078163309e-08
ii= 519 , relres= 1.1974543273816298e-07 , err= 4.044367518747993e-08
ii= 520 , relres= 1.1328386581881848e-07 , err= 3.861627521340705e-08
ii= 521 , relres= 9.771152600992724e-08 , err= 3.7841996046192056e-08
ii= 522 , relres= 9.546263086956309e-08 , err= 3.711927863592387e-08
ii= 523 , relres= 1.0402773241094384e-07 , err= 3.562075445709452e-08
ii= 524 , relres= 9.486732213624172e-08 , err= 3.430294925135436e-08
ii= 525 , relres= 9.528849703070176e-08 , err= 3.3973268318815726e-08
ii= 526 , relres= 8.951990598383064e-08 , err= 3.2315085634039133e-08
ii= 527 , relres= 8.168772846207592e-08 , err= 3.1477182882209825e-08
ii= 528 , relres= 8.454913024448009e-08 , err= 2.9560874518068514e-08
ii= 529 , relres= 7.548148522253981e-08 , err= 2.8451363387972262e-08
ii= 530 , relres= 7.417223645512661e-08 , err= 2.7957236585891197e-08
ii= 531 , relres= 7.150252463879385e-08 , err= 2.7133920577283458e-08
ii= 532 , relres= 7.679534771401339e-08 , err= 2.492846064717895e-08
ii= 533 , relres= 7.12208683969027e-08 , err= 2.4407772685004378e-08
ii= 534 , relres= 6.632281809523871e-08 , err= 2.3912754965850608e-08
ii= 535 , relres= 6.767454709042764e-08 , err= 2.3405770713340672e-08
ii= 536 , relres= 6.334881326144633e-08 , err= 2.2813670237611157e-08
ii= 537 , relres= 5.98459212263423e-08 , err= 2.2298977449008333e-08
ii= 538 , relres= 6.029018811044579e-08 , err= 2.2047300050345893e-08
ii= 539 , relres= 5.492030273938661e-08 , err= 1.9814145332899914e-08
ii= 540 , relres= 5.088139380248511e-08 , err= 1.935783343533824e-08
ii= 541 , relres= 5.009137044713507e-08 , err= 1.888759240928234e-08
ii= 542 , relres= 6.034320835672006e-08 , err= 1.7904139733045577e-08
ii= 543 , relres= 4.488866064318086e-08 , err= 1.659499962970725e-08
ii= 544 , relres= 4.592656913241546e-08 , err= 1.639290164191354e-08
ii= 545 , relres= 4.726466762213827e-08 , err= 1.6001738146682597e-08
ii= 546 , relres= 4.210726646163057e-08 , err= 1.5530225574998032e-08
ii= 547 , relres= 4.23786490945952e-08 , err= 1.5229987525333977e-08
ii= 548 , relres= 4.3440935602547264e-08 , err= 1.4752228546351941e-08
ii= 549 , relres= 4.163405713657033e-08 , err= 1.398216928155477e-08
ii= 550 , relres= 3.785984509815731e-08 , err= 1.366300900831793e-08

```

```

ii= 551 , relres= 3.6532380017872244e-08 , err= 1.3105311303504267e-08
ii= 552 , relres= 3.624118114182196e-08 , err= 1.26485485983641e-08
ii= 553 , relres= 3.768203637068236e-08 , err= 1.237349621024553e-08
ii= 554 , relres= 3.278976046994161e-08 , err= 1.1899448495475511e-08
ii= 555 , relres= 3.031946610380481e-08 , err= 1.1570030140179254e-08
ii= 556 , relres= 3.465485663891747e-08 , err= 1.1089762212951479e-08
ii= 557 , relres= 3.276839448712025e-08 , err= 1.0677730518077997e-08
ii= 558 , relres= 3.117351438700295e-08 , err= 1.0403273627667934e-08
ii= 559 , relres= 2.9351945774798893e-08 , err= 9.966587214586197e-09
ii= 560 , relres= 2.8021965395909413e-08 , err= 9.846959516502641e-09
ii= 561 , relres= 2.565940513995669e-08 , err= 9.318517021834586e-09
ii= 562 , relres= 2.4544783356441863e-08 , err= 8.82213062068597e-09
ii= 563 , relres= 2.5245855490778526e-08 , err= 8.462986071102928e-09
ii= 564 , relres= 2.3124309904853406e-08 , err= 8.221885628308679e-09
ii= 565 , relres= 2.0207768785332166e-08 , err= 8.118642455910608e-09
ii= 566 , relres= 2.01164470317877e-08 , err= 8.013607704180022e-09
ii= 567 , relres= 2.2946612252696887e-08 , err= 7.330018687593229e-09
ii= 568 , relres= 1.9837757939382138e-08 , err= 7.139015127507767e-09
ii= 569 , relres= 1.8812058266022693e-08 , err= 7.003460348181818e-09
ii= 570 , relres= 1.864826621835937e-08 , err= 6.841234986302977e-09
ii= 571 , relres= 1.8679860999408832e-08 , err= 6.737802118194568e-09
ii= 572 , relres= 1.8703749116758015e-08 , err= 6.570933081494478e-09
ii= 573 , relres= 1.743696687340251e-08 , err= 6.341664990902972e-09
ii= 574 , relres= 1.7718815320302014e-08 , err= 6.1196128575384e-09
ii= 575 , relres= 1.5626393539181943e-08 , err= 5.90417234094611e-09
ii= 576 , relres= 1.531793220934596e-08 , err= 5.7521298738646644e-09
ii= 577 , relres= 1.5572852606903827e-08 , err= 5.629277522402011e-09
ii= 578 , relres= 1.617076135944266e-08 , err= 5.420258681647614e-09
ii= 579 , relres= 1.6354095544962737e-08 , err= 5.198097572039853e-09
ii= 580 , relres= 1.409882142738497e-08 , err= 5.0576850528425675e-09
ii= 581 , relres= 1.3780985784557324e-08 , err= 4.828688157316223e-09
ii= 582 , relres= 1.2791207641725358e-08 , err= 4.763624616943434e-09
ii= 583 , relres= 1.2381383869151793e-08 , err= 4.607988333543675e-09
ii= 584 , relres= 1.2400919239670405e-08 , err= 4.356743505227683e-09
ii= 585 , relres= 1.197245190105905e-08 , err= 4.23721556114465e-09
ii= 586 , relres= 1.170486823558564e-08 , err= 4.175846933235695e-09
ii= 587 , relres= 1.067068831666227e-08 , err= 4.025884598695703e-09
ii= 588 , relres= 1.0395531237909686e-08 , err= 3.946032325991091e-09
ii= 589 , relres= 1.0942542635512851e-08 , err= 3.863043092880143e-09
ii= 590 , relres= 1.200705137408901e-08 , err= 3.6372921511197195e-09
ii= 591 , relres= 9.886865659472784e-09 , err= 3.5104420497458905e-09
ii= 592 , relres= 9.259521707571627e-09 , err= 3.4185387577288333e-09
ii= 593 , relres= 9.487819776906857e-09 , err= 3.324340252490706e-09
ii= 594 , relres= 8.962012664180469e-09 , err= 3.1788512863735788e-09
ii= 595 , relres= 9.293902458452338e-09 , err= 3.0561011376073478e-09
ii= 596 , relres= 8.747431339074077e-09 , err= 2.952280633861753e-09
ii= 597 , relres= 8.400868559804612e-09 , err= 2.8426266149667875e-09
ii= 598 , relres= 7.48181506296106e-09 , err= 2.7668192008342165e-09
ii= 599 , relres= 7.471391835865748e-09 , err= 2.683647957435927e-09
ii= 600 , relres= 7.248708512687688e-09 , err= 2.6267154978165513e-09
ii= 601 , relres= 8.173016239771451e-09 , err= 2.5400435174379324e-09
ii= 602 , relres= 6.584843102909193e-09 , err= 2.4103624831479308e-09
ii= 603 , relres= 6.754412109299783e-09 , err= 2.377100695916839e-09
ii= 604 , relres= 6.193995968551684e-09 , err= 2.283495930660395e-09
ii= 605 , relres= 5.8661995244603e-09 , err= 2.2477716978482022e-09
ii= 606 , relres= 5.931823800528014e-09 , err= 2.1805313042735733e-09
ii= 607 , relres= 5.871096080241152e-09 , err= 2.154867538381889e-09
ii= 608 , relres= 5.976787965799236e-09 , err= 2.0703290220198307e-09
ii= 609 , relres= 5.77134448950617e-09 , err= 2.0101111622331705e-09
ii= 610 , relres= 5.435215307446774e-09 , err= 1.9646338669038503e-09
ii= 611 , relres= 4.952348667326607e-09 , err= 1.938047815204357e-09

```

```

ii= 612 , relres= 5.200138352167244e-09 , err= 1.8757054629469082e-09
ii= 613 , relres= 4.796785311607848e-09 , err= 1.75533414164031e-09
ii= 614 , relres= 4.518496081164644e-09 , err= 1.664090790556933e-09
ii= 615 , relres= 4.411058511048728e-09 , err= 1.630530504198883e-09
ii= 616 , relres= 4.540945815208987e-09 , err= 1.5837769145425634e-09
ii= 617 , relres= 3.9993985907577154e-09 , err= 1.488357373165692e-09
ii= 618 , relres= 3.797648355594239e-09 , err= 1.407873300842689e-09
ii= 619 , relres= 3.6179946544731893e-09 , err= 1.359886410477818e-09
ii= 620 , relres= 3.45098809016736e-09 , err= 1.3383799042692372e-09
ii= 621 , relres= 3.460374056493314e-09 , err= 1.3090610957899133e-09
ii= 622 , relres= 3.6532903792903e-09 , err= 1.2768812221813324e-09
ii= 623 , relres= 3.5166525200121785e-09 , err= 1.2468408046086575e-09
ii= 624 , relres= 3.284290939169435e-09 , err= 1.200080616932111e-09
ii= 625 , relres= 3.2250193941064787e-09 , err= 1.1830390181043308e-09
ii= 626 , relres= 2.9976705174277515e-09 , err= 1.158815208673766e-09
ii= 627 , relres= 3.110801758450809e-09 , err= 1.1378557566975023e-09
ii= 628 , relres= 3.003939615863061e-09 , err= 1.1057686673496764e-09
ii= 629 , relres= 3.277767429129384e-09 , err= 1.0514641878879166e-09
ii= 630 , relres= 3.374798317081596e-09 , err= 1.0036328472477201e-09
ii= 631 , relres= 2.9567599343717305e-09 , err= 9.575836078937397e-10
ii= 632 , relres= 2.5168593094344764e-09 , err= 9.246445978973661e-10
ii= 633 , relres= 2.421558020187273e-09 , err= 8.588987339632468e-10
ii= 634 , relres= 2.416351785891021e-09 , err= 8.417007381463381e-10
ii= 635 , relres= 2.32448678499986e-09 , err= 7.913167152663637e-10
ii= 636 , relres= 2.0545866468004375e-09 , err= 7.570392312556986e-10
ii= 637 , relres= 2.039548821416856e-09 , err= 7.468667852125068e-10
ii= 638 , relres= 2.020560673459134e-09 , err= 7.215035602577098e-10
ii= 639 , relres= 2.0538615042078937e-09 , err= 6.923910371635594e-10
ii= 640 , relres= 1.8890469363622956e-09 , err= 6.613820440621451e-10
ii= 641 , relres= 1.8837368826644353e-09 , err= 6.435481646008422e-10
ii= 642 , relres= 1.6732599835496735e-09 , err= 6.333958303252101e-10
ii= 643 , relres= 1.771337856992386e-09 , err= 5.984132468749566e-10
ii= 644 , relres= 1.5859220952373082e-09 , err= 5.844943623536829e-10
ii= 645 , relres= 1.6294566765486592e-09 , err= 5.764982432532434e-10
ii= 646 , relres= 1.6192127473247402e-09 , err= 5.54402261632664e-10
ii= 647 , relres= 1.4292037455105506e-09 , err= 5.404866491882493e-10
ii= 648 , relres= 1.4191176288043823e-09 , err= 5.353257274883294e-10
ii= 649 , relres= 1.4518568360626447e-09 , err= 5.225905305481021e-10
ii= 650 , relres= 1.3907178609485527e-09 , err= 5.125008051970517e-10
ii= 651 , relres= 1.3886085191551542e-09 , err= 5.044144234985368e-10
ii= 652 , relres= 1.4781728632031688e-09 , err= 4.910229347864009e-10
ii= 653 , relres= 1.3754092561409542e-09 , err= 4.600199922585228e-10
ii= 654 , relres= 1.2414350882871046e-09 , err= 4.4783015243516083e-10
ii= 655 , relres= 1.2406869852352002e-09 , err= 4.3881515791388326e-10
ii= 656 , relres= 1.1118367038644715e-09 , err= 4.098670770376188e-10
ii= 657 , relres= 1.1189171176227041e-09 , err= 3.96930067721749e-10
ii= 658 , relres= 1.069027136794695e-09 , err= 3.870578395617889e-10
ii= 659 , relres= 1.0635772137444615e-09 , err= 3.7399717202360303e-10
ii= 660 , relres= 1.002206850335747e-09 , err= 3.6366304337948756e-10
ii= 661 , relres= 1.029730756450997e-09 , err= 3.5571691710890824e-10
ii= 662 , relres= 9.955526405153967e-10 , err= 3.483114347293007e-10
ii= 663 , relres= 9.290009027129453e-10 , err= 3.28623928228577e-10
ii= 664 , relres= 8.623703874371884e-10 , err= 3.1710898644019773e-10
ii= 665 , relres= 8.242117989237318e-10 , err= 3.113230189833549e-10
ii= 666 , relres= 9.460162797926477e-10 , err= 3.009347886925713e-10
ii= 667 , relres= 8.60749027677281e-10 , err= 2.918780914536598e-10
ii= 668 , relres= 8.823304623496594e-10 , err= 2.8522921693779526e-10
ii= 669 , relres= 7.841027853082386e-10 , err= 2.6619452400582546e-10
ii= 670 , relres= 7.733081564190666e-10 , err= 2.637806430781847e-10
ii= 671 , relres= 7.489847388672677e-10 , err= 2.49564976881613e-10
ii= 672 , relres= 6.98725777181255e-10 , err= 2.4609839936868616e-10

```

```

ii= 673 , relres= 7.184984496465112e-10 , err= 2.296606057093519e-10
ii= 674 , relres= 7.197560712128873e-10 , err= 2.1856279348294883e-10
ii= 675 , relres= 6.542190177667628e-10 , err= 2.111919250008411e-10
ii= 676 , relres= 5.720929905850793e-10 , err= 2.0638479532510783e-10
ii= 677 , relres= 5.607536731943496e-10 , err= 2.0331554632506156e-10
ii= 678 , relres= 5.887463621827777e-10 , err= 1.918852440072817e-10
ii= 679 , relres= 5.697709373298194e-10 , err= 1.865348198399438e-10
ii= 680 , relres= 5.952999472552405e-10 , err= 1.8059979835192107e-10
ii= 681 , relres= 5.366058790973222e-10 , err= 1.7518089561661725e-10
ii= 682 , relres= 4.868215042587287e-10 , err= 1.6844678633487483e-10
ii= 683 , relres= 4.751272427070469e-10 , err= 1.6182977492133182e-10
ii= 684 , relres= 4.4655528513687884e-10 , err= 1.58085794431308e-10
ii= 685 , relres= 4.678168430915056e-10 , err= 1.5120017337415638e-10
ii= 686 , relres= 4.132180467597938e-10 , err= 1.4206125910965208e-10
ii= 687 , relres= 3.815193541557697e-10 , err= 1.4114101359643351e-10
ii= 688 , relres= 3.98173707589946e-10 , err= 1.3540527535047833e-10
ii= 689 , relres= 4.1151606778343333e-10 , err= 1.3294632688137413e-10
ii= 690 , relres= 3.9598319870882514e-10 , err= 1.2978236111489534e-10
ii= 691 , relres= 3.82085364576824e-10 , err= 1.252841643187095e-10
ii= 692 , relres= 3.540766304622166e-10 , err= 1.2211383596490254e-10
ii= 693 , relres= 3.7022443511390385e-10 , err= 1.1981902549748592e-10
ii= 694 , relres= 3.350022338331364e-10 , err= 1.156707267669611e-10
ii= 695 , relres= 3.014470716037006e-10 , err= 1.1222122049678576e-10
ii= 696 , relres= 3.0082124678343425e-10 , err= 1.0961958812284892e-10
ii= 697 , relres= 3.193661650160713e-10 , err= 1.0686864446555613e-10
ii= 698 , relres= 3.3057107320086294e-10 , err= 1.0262563676312004e-10
ii= 699 , relres= 2.7846592199743727e-10 , err= 1.0013289158134875e-10
ii= 700 , relres= 2.715726468458168e-10 , err= 9.793924139235223e-11
ii= 701 , relres= 2.6784749269254997e-10 , err= 9.620932157281822e-11
ii= 702 , relres= 2.839661227076541e-10 , err= 9.312210490881048e-11
ii= 703 , relres= 2.748679843460147e-10 , err= 9.097638247308906e-11
ii= 704 , relres= 2.760573348810781e-10 , err= 8.852538014441608e-11
ii= 705 , relres= 2.3699978011606126e-10 , err= 8.622553895400484e-11
ii= 706 , relres= 2.656122836076207e-10 , err= 8.271129314740402e-11
ii= 707 , relres= 2.2484606413852003e-10 , err= 8.020396532116993e-11
ii= 708 , relres= 2.356095358629733e-10 , err= 7.781867111698913e-11
ii= 709 , relres= 2.4185011942196553e-10 , err= 7.697057916098866e-11
ii= 710 , relres= 2.3190091585359418e-10 , err= 7.229459297569928e-11
ii= 711 , relres= 2.1975229691540025e-10 , err= 6.961510432921757e-11
ii= 712 , relres= 2.1253734661112768e-10 , err= 6.734464233066796e-11
ii= 713 , relres= 1.9397656598377323e-10 , err= 6.582895152281709e-11
ii= 714 , relres= 2.0360471850918586e-10 , err= 6.377068020165266e-11
ii= 715 , relres= 2.0687004346427035e-10 , err= 6.2024386945457e-11
ii= 716 , relres= 1.9286247787660682e-10 , err= 5.880521714442045e-11
ii= 717 , relres= 1.9367371535021206e-10 , err= 5.738310110043979e-11
ii= 718 , relres= 2.0141244611300733e-10 , err= 5.5936749180973315e-11
ii= 719 , relres= 1.8110681113868542e-10 , err= 5.344384920262145e-11
ii= 720 , relres= 1.7039778671519683e-10 , err= 5.139489411001569e-11
ii= 721 , relres= 1.7562880566051563e-10 , err= 5.0064522734753033e-11
ii= 722 , relres= 1.6884494124441074e-10 , err= 4.894985291216536e-11
ii= 723 , relres= 1.652633569623295e-10 , err= 4.625447287170291e-11
ii= 724 , relres= 1.2782199267867687e-10 , err= 4.3629118079959366e-11
ii= 725 , relres= 1.3710794817475223e-10 , err= 4.242276630468066e-11
ii= 726 , relres= 1.6076466505766557e-10 , err= 4.1267726833766304e-11
ii= 727 , relres= 1.3121119903156726e-10 , err= 3.890028330960766e-11
ii= 728 , relres= 1.6362680383898828e-10 , err= 3.799373319977624e-11
ii= 729 , relres= 1.6503704177104503e-10 , err= 3.725927529735939e-11
ii= 730 , relres= 2.0041827080152877e-10 , err= 3.653463560110366e-11
ii= 731 , relres= 1.425850815562881e-10 , err= 3.532731190952669e-11
ii= 732 , relres= 1.4439221135516553e-10 , err= 3.479079244202593e-11
ii= 733 , relres= 1.5641703492718623e-10 , err= 3.2582425638937017e-11

```

```
ii= 734 , relres= 1.2299420309352994e-10 , err= 3.205240999492252e-11
ii= 735 , relres= 1.3318711420234713e-10 , err= 3.165541816767073e-11
ii= 736 , relres= 1.187595417283085e-10 , err= 3.1078544027234615e-11
ii= 737 , relres= 1.146724259781195e-10 , err= 3.067899785376668e-11
ii= 738 , relres= 1.1375629601635629e-10 , err= 3.0284162687207536e-11
ii= 739 , relres= 1.2616900166494258e-10 , err= 2.998823077185579e-11
ii= 740 , relres= 1.1463171046827592e-10 , err= 2.972583935243186e-11
ii= 741 , relres= 1.2774573622061607e-10 , err= 2.945527146711186e-11
ii= 742 , relres= 1.1331921774814827e-10 , err= 2.922104977645716e-11
ii= 743 , relres= 1.2116590898056727e-10 , err= 2.905514682502227e-11
ii= 744 , relres= 1.1829931458551693e-10 , err= 2.8907592359567628e-11
ii= 745 , relres= 1.0917507694002097e-10 , err= 2.8742203383057268e-11
ii= 746 , relres= 1.1466522494883563e-10 , err= 2.8581306840201037e-11
ii= 747 , relres= 1.0373138611235642e-10 , err= 2.842224656586631e-11
ii= 748 , relres= 1.332275216885091e-10 , err= 2.818899325067139e-11
ii= 749 , relres= 1.2852168573303132e-10 , err= 2.7850017853394237e-11
ii= 750 , relres= 1.1197729264585059e-10 , err= 2.7557843130368814e-11
ii= 751 , relres= 1.291275057332542e-10 , err= 2.7343110080019082e-11
ii= 752 , relres= 1.1378349689405847e-10 , err= 2.7135813527432504e-11
ii= 753 , relres= 1.1862123779031047e-10 , err= 2.691626075971643e-11
ii= 754 , relres= 1.2766783189037112e-10 , err= 2.6720765948866165e-11
ii= 755 , relres= 1.1921504763010452e-10 , err= 2.6580968689610447e-11
ii= 756 , relres= 1.3285557714383652e-10 , err= 2.6477848377124162e-11
ii= 757 , relres= 1.0830514629960468e-10 , err= 2.637584410289874e-11
ii= 758 , relres= 1.2743888240619005e-10 , err= 2.6247115056897384e-11
ii= 759 , relres= 1.2403743145092155e-10 , err= 2.6090705539404565e-11
ii= 760 , relres= 1.211314109172528e-10 , err= 2.5936218733877064e-11
ii= 761 , relres= 1.5066176510328563e-10 , err= 2.579898020497055e-11
ii= 762 , relres= 1.1639808369232615e-10 , err= 2.5684572303806755e-11
ii= 763 , relres= 1.2037265012167347e-10 , err= 2.5578851564026558e-11
ii= 764 , relres= 1.1663161324194028e-10 , err= 2.547203232393262e-11
ii= 765 , relres= 1.2345245366385384e-10 , err= 2.5360254916086768e-11
ii= 766 , relres= 1.2791725754711811e-10 , err= 2.5246182950862063e-11
ii= 767 , relres= 1.3518814048424315e-10 , err= 2.5136111375806393e-11
ii= 768 , relres= 1.4419530237579577e-10 , err= 2.503456642620427e-11
ii= 769 , relres= 1.5293087026294793e-10 , err= 2.4941352284279583e-11
ii= 770 , relres= 1.3724074988212287e-10 , err= 2.4852842839468616e-11
ii= 771 , relres= 1.4998427950796288e-10 , err= 2.4764942489102665e-11
ii= 772 , relres= 1.56487366556872e-10 , err= 2.467584689952367e-11
ii= 773 , relres= 1.335311285837779e-10 , err= 2.4587618604054427e-11
ii= 774 , relres= 1.3186010172421635e-10 , err= 2.4505027310269263e-11
ii= 775 , relres= 1.549127024542274e-10 , err= 2.443177355294595e-11
ii= 776 , relres= 1.3313976808282817e-10 , err= 2.436767817096064e-11
ii= 777 , relres= 1.2139502031982218e-10 , err= 2.4308968150106873e-11
ii= 778 , relres= 1.2385458348654485e-10 , err= 2.4249956059002283e-11
ii= 779 , relres= 1.2871823772514608e-10 , err= 2.4184412868811034e-11
ii= 780 , relres= 1.363751294973472e-10 , err= 2.4107267351014578e-11
ii= 781 , relres= 1.534252598907705e-10 , err= 2.401781213117215e-11
ii= 782 , relres= 1.3885727936636128e-10 , err= 2.3922426464304108e-11
ii= 783 , relres= 1.5854952747505345e-10 , err= 2.3831703651668265e-11
ii= 784 , relres= 1.4580540298251488e-10 , err= 2.3752692874314054e-11
ii= 785 , relres= 1.4122146241686711e-10 , err= 2.3685036462981327e-11
ii= 786 , relres= 1.3211929732828925e-10 , err= 2.362325466534717e-11
ii= 787 , relres= 1.2133156209906568e-10 , err= 2.3559903033840928e-11
ii= 788 , relres= 1.42549456629474e-10 , err= 2.348728288877071e-11
ii= 789 , relres= 1.5209032096647252e-10 , err= 2.3399389602942493e-11
ii= 790 , relres= 1.381157668619506e-10 , err= 2.3295920453721723e-11
ii= 791 , relres= 1.3078392358427642e-10 , err= 2.318547293131146e-11
ii= 792 , relres= 1.6991696164154804e-10 , err= 2.3081209798701782e-11
ii= 793 , relres= 1.2346234754054405e-10 , err= 2.2991232934629823e-11
ii= 794 , relres= 1.2405521537972726e-10 , err= 2.2914799504063702e-11
```

```
ii= 795 , relres= 1.4014438411082273e-10 , err= 2.2845748849864763e-11
ii= 796 , relres= 1.566895348102856e-10 , err= 2.277643744205771e-11
ii= 797 , relres= 1.7952440699284278e-10 , err= 2.2699729028434206e-11
ii= 798 , relres= 1.5743538703258725e-10 , err= 2.261068222956576e-11
ii= 799 , relres= 1.477566168764686e-10 , err= 2.2509175481963114e-11
ii= 800 , relres= 1.6254701780840224e-10 , err= 2.240141821464577e-11
ii= 801 , relres= 1.3973664333963297e-10 , err= 2.2296905396141852e-11
ii= 802 , relres= 1.4395200587458992e-10 , err= 2.2202480348097663e-11
ii= 803 , relres= 1.3957496871913484e-10 , err= 2.2119351998648866e-11
ii= 804 , relres= 1.372164231030195e-10 , err= 2.204460075545217e-11
ii= 805 , relres= 1.338128826835229e-10 , err= 2.197393625512189e-11
ii= 806 , relres= 1.654725561322258e-10 , err= 2.1903707191385322e-11
ii= 807 , relres= 1.3265482209865414e-10 , err= 2.1831956730945333e-11
ii= 808 , relres= 1.083553801063076e-10 , err= 2.1758802413026803e-11
ii= 809 , relres= 1.3948061846325105e-10 , err= 2.1685829134037876e-11
ii= 810 , relres= 1.3051406302788658e-10 , err= 2.161476813312362e-11
ii= 811 , relres= 1.4214174922763587e-10 , err= 2.1546314546155884e-11
ii= 812 , relres= 1.2300533209914519e-10 , err= 2.1479892696243178e-11
ii= 813 , relres= 1.3837915201110794e-10 , err= 2.1414207645870053e-11
ii= 814 , relres= 1.5721190629783638e-10 , err= 2.1347948519704464e-11
ii= 815 , relres= 1.3035313385349802e-10 , err= 2.128034794638689e-11
ii= 816 , relres= 1.0610306905546267e-10 , err= 2.121347666957903e-11
ii= 817 , relres= 1.5492247219169712e-10 , err= 2.1147715492704308e-11
ii= 818 , relres= 1.4672422641576678e-10 , err= 2.1083594138595175e-11
ii= 819 , relres= 1.3683068612550109e-10 , err= 2.1021735007891457e-11
ii= 820 , relres= 1.6078641350705025e-10 , err= 2.0962712359980582e-11
ii= 821 , relres= 1.4590553754313924e-10 , err= 2.0906926427375008e-11
ii= 822 , relres= 1.2190424311101057e-10 , err= 2.0854498442029302e-11
ii= 823 , relres= 1.4415957397665554e-10 , err= 2.0805246162964296e-11
ii= 824 , relres= 1.3383349297249742e-10 , err= 2.0758679317748502e-11
ii= 825 , relres= 1.1966154359643907e-10 , err= 2.071410871974295e-11
ii= 826 , relres= 1.235621976187278e-10 , err= 2.0670782334780732e-11
ii= 827 , relres= 1.2796962979116645e-10 , err= 2.062800022748584e-11
ii= 828 , relres= 1.1409201491312087e-10 , err= 2.058527383703732e-11
ii= 829 , relres= 1.3352595629541856e-10 , err= 2.054241856382907e-11
ii= 830 , relres= 1.3456318931300636e-10 , err= 2.0499581542824608e-11
ii= 831 , relres= 1.225923780926733e-10 , err= 2.04571763850124e-11
ii= 832 , relres= 1.229309076224109e-10 , err= 2.0415728536102862e-11
ii= 833 , relres= 1.2666707605169474e-10 , err= 2.037565158666707e-11
ii= 834 , relres= 1.2099777269136118e-10 , err= 2.0337061087345908e-11
ii= 835 , relres= 1.2944542904457394e-10 , err= 2.0299660916106546e-11
ii= 836 , relres= 1.2748013481916885e-10 , err= 2.0262802705772556e-11
ii= 837 , relres= 1.1303073998315215e-10 , err= 2.0225512350292628e-11
ii= 838 , relres= 1.1403947598598493e-10 , err= 2.0186648851123973e-11
ii= 839 , relres= 1.1715994337643166e-10 , err= 2.0145069585826742e-11
ii= 840 , relres= 1.2320577129868668e-10 , err= 2.0099885465218838e-11
ii= 841 , relres= 1.1740136167121928e-10 , err= 2.0050795838911657e-11
ii= 842 , relres= 1.141956255906484e-10 , err= 1.9998447037704686e-11
ii= 843 , relres= 1.0815335927853821e-10 , err= 1.9944562400112928e-11
ii= 844 , relres= 1.2093604982503652e-10 , err= 1.989154236386328e-11
ii= 845 , relres= 1.211532957176343e-10 , err= 1.9841669727891804e-11
ii= 846 , relres= 1.3540336914966907e-10 , err= 1.979631326758784e-11
ii= 847 , relres= 1.2114365416524883e-10 , err= 1.9755621596052113e-11
ii= 848 , relres= 1.4552282220122651e-10 , err= 1.971870620204216e-11
ii= 849 , relres= 1.157875995933786e-10 , err= 1.9684080910310102e-11
ii= 850 , relres= 1.2386611883837059e-10 , err= 1.9649993015355368e-11
ii= 851 , relres= 1.2215364012184184e-10 , err= 1.9614678527336213e-11
ii= 852 , relres= 1.2805734269420474e-10 , err= 1.9576511530241656e-11
ii= 853 , relres= 1.4263247742012687e-10 , err= 1.9534275422497108e-11
ii= 854 , relres= 1.2774511356376095e-10 , err= 1.948748718152216e-11
ii= 855 , relres= 1.417840075514237e-10 , err= 1.9436715461400777e-11
```

```

ii= 856 , relres= 1.1154202817286756e-10 , err= 1.9383547723937716e-11
ii= 857 , relres= 1.3825586151653537e-10 , err= 1.9330074647129004e-11
ii= 858 , relres= 1.094832291249541e-10 , err= 1.927807386744714e-11
ii= 859 , relres= 1.1676338776259892e-10 , err= 1.922841321387393e-11
ii= 860 , relres= 1.156928758698012e-10 , err= 1.918094984059831e-11
ii= 861 , relres= 1.1325006601863484e-10 , err= 1.9134805093129578e-11
ii= 862 , relres= 1.1346221726124316e-10 , err= 1.9088770768935853e-11
ii= 863 , relres= 1.2953780403709409e-10 , err= 1.9041737968295296e-11
ii= 864 , relres= 1.0677014483883616e-10 , err= 1.8993098066460833e-11
ii= 865 , relres= 1.1409343691657699e-10 , err= 1.894314237609285e-11
ii= 866 , relres= 1.2191222428639168e-10 , err= 1.8893189475147344e-11
ii= 867 , relres= 1.3349992990989305e-10 , err= 1.8845241977948378e-11
ii= 868 , relres= 1.1035106831172446e-10 , err= 1.8801187383895378e-11
ii= 869 , relres= 1.6015117288074822e-10 , err= 1.8762032022161476e-11
ii= 870 , relres= 1.1809641494665328e-10 , err= 1.8727634117629866e-11
ii= 871 , relres= 1.0584133917478273e-10 , err= 1.869690566686401e-11
ii= 872 , relres= 1.1364308575102995e-10 , err= 1.8668270856586914e-11
ii= 873 , relres= 1.052261854755132e-10 , err= 1.8640008757924448e-11
ii= 874 , relres= 1.2207353282992147e-10 , err= 1.8610452702975366e-11
ii= 875 , relres= 1.1081964908357456e-10 , err= 1.857815070714853e-11
ii= 876 , relres= 1.1852267646717498e-10 , err= 1.8542136874856628e-11
ii= 877 , relres= 1.129972484947277e-10 , err= 1.8502293600782296e-11
ii= 878 , relres= 1.300258757418055e-10 , err= 1.845959091180202e-11
ii= 879 , relres= 1.477629722435425e-10 , err= 1.8415872295412593e-11
ii= 880 , relres= 1.0996336628180683e-10 , err= 1.8373135740150734e-11
ii= 881 , relres= 1.164299805353699e-10 , err= 1.8332762766274452e-11
ii= 882 , relres= 1.1991879994789002e-10 , err= 1.8295101041761292e-11
ii= 883 , relres= 1.1719632320695247e-10 , err= 1.8259644415600502e-11
ii= 884 , relres= 1.151439125357118e-10 , err= 1.8225369734851976e-11
ii= 885 , relres= 1.164845405090363e-10 , err= 1.8191110612880254e-11
ii= 886 , relres= 1.0143562603851966e-10 , err= 1.8155893077353552e-11
ii= 887 , relres= 1.073262013781763e-10 , err= 1.81192139083728e-11
ii= 888 , relres= 1.3223927847904894e-10 , err= 1.8081356322438397e-11
ii= 889 , relres= 1.1576342711290538e-10 , err= 1.8043423575586816e-11
ii= 890 , relres= 1.1047217935533401e-10 , err= 1.800704576598606e-11
ii= 891 , relres= 1.055796941656416e-10 , err= 1.7973723059437678e-11
ii= 892 , relres= 1.0373161904984687e-10 , err= 1.7943104895996353e-11
ii= 893 , relres= 1.0484724383815908e-10 , err= 1.7915128819982425e-11
ii= 894 , relres= 1.1638056871740438e-10 , err= 1.7889284549409957e-11
ii= 895 , relres= 1.1784195961939062e-10 , err= 1.786482161649936e-11
ii= 896 , relres= 1.023282978877676e-10 , err= 1.7840961971222657e-11
ii= 897 , relres= 1.0956215230313413e-10 , err= 1.7817082925872745e-11
ii= 898 , relres= 1.0652756329626937e-10 , err= 1.779281613977034e-11
ii= 899 , relres= 1.0107954002911188e-10 , err= 1.7768102307229523e-11
ii= 900 , relres= 1.0820216938556407e-10 , err= 1.7743104079429107e-11
ii= 901 , relres= 1.1259366659224629e-10 , err= 1.7718039463843438e-11
ii= 902 , relres= 1.0778962661355233e-10 , err= 1.7692939501166786e-11
ii= 903 , relres= 1.0283983365227658e-10 , err= 1.7667535768669566e-11
ii= 904 , relres= 1.2227476820685088e-10 , err= 1.7641599554518704e-11
ii= 905 , relres= 1.0700618991206862e-10 , err= 1.7614443742816307e-11
ii= 906 , relres= 1.026035487875692e-10 , err= 1.7585429847753327e-11
ii= 907 , relres= 1.2600437186311272e-10 , err= 1.7554196276431258e-11
ii= 908 , relres= 1.2357932633510647e-10 , err= 1.7520942782879186e-11
ii= 909 , relres= 1.214994417962975e-10 , err= 1.7486666559612923e-11
ii= 910 , relres= 1.0592130399672501e-10 , err= 1.7453093512550017e-11
ii= 911 , relres= 1.0316223975996093e-10 , err= 1.7422167550618148e-11
ii= 912 , relres= 1.107042339894264e-10 , err= 1.7395272184537235e-11
ii= 913 , relres= 1.1296090474351103e-10 , err= 1.737273272212161e-11
ii= 914 , relres= 1.133604959280449e-10 , err= 1.735388465022333e-11
ii= 915 , relres= 1.1339533785449208e-10 , err= 1.7337541636185166e-11
ii= 916 , relres= 1.253583014442668e-10 , err= 1.7322386304603175e-11

```

```

ii= 917 , relres= 1.0495140138387596e-10 , err= 1.730905565536016e-11
ii= 918 , relres= 1.0804490986112847e-10 , err= 1.7296817796225e-11
ii= 919 , relres= 1.0310535054656076e-10 , err= 1.7285116232752896e-11
ii= 920 , relres= 1.1432267174834983e-10 , err= 1.7273538238199355e-11
ii= 921 , relres= 1.0889983511454352e-10 , err= 1.7261795892762874e-11
ii= 922 , relres= 1.2265476615778766e-10 , err= 1.7249727033829247e-11
ii= 923 , relres= 1.199581353952903e-10 , err= 1.7237288678865195e-11
ii= 924 , relres= 1.2342305928661466e-10 , err= 1.722451387470133e-11
ii= 925 , relres= 1.2386399013746908e-10 , err= 1.7211499325794675e-11
ii= 926 , relres= 1.2155939429638555e-10 , err= 1.7198359484612595e-11
ii= 927 , relres= 1.4216323055711573e-10 , err= 1.7185158032369814e-11
ii= 928 , relres= 1.3248694370609758e-10 , err= 1.7171911474442477e-11
ii= 929 , relres= 1.176028265412123e-10 , err= 1.7158575727632796e-11
ii= 930 , relres= 1.3644513887980528e-10 , err= 1.7145053857596087e-11
ii= 931 , relres= 1.3223644555866252e-10 , err= 1.713121180470103e-11
ii= 932 , relres= 1.3026093557934732e-10 , err= 1.711691093158172e-11
ii= 933 , relres= 1.238865194551141e-10 , err= 1.710200761808454e-11
ii= 934 , relres= 1.2624990050743335e-10 , err= 1.7086386829820476e-11
ii= 935 , relres= 1.2705707051510496e-10 , err= 1.7069965269344972e-11
ii= 936 , relres= 1.1786006976457263e-10 , err= 1.7052681150125825e-11
ii= 937 , relres= 1.1537330249076786e-10 , err= 1.7034520417081215e-11
ii= 938 , relres= 1.2588416402820746e-10 , err= 1.701551720841068e-11
ii= 939 , relres= 1.2735219778778606e-10 , err= 1.6995742096413678e-11
ii= 940 , relres= 1.2563129033634753e-10 , err= 1.6975296355588787e-11
ii= 941 , relres= 1.1634467313589713e-10 , err= 1.6954291026864434e-11
ii= 942 , relres= 1.0877633648784065e-10 , err= 1.693285695279162e-11
ii= 943 , relres= 1.1366428142842606e-10 , err= 1.6911121850433332e-11
ii= 944 , relres= 1.1522401306361535e-10 , err= 1.688921508138825e-11
ii= 945 , relres= 1.0858417659346001e-10 , err= 1.686727379124757e-11
ii= 946 , relres= 1.1931265399116842e-10 , err= 1.6845463448223247e-11
ii= 947 , relres= 1.1425890405789053e-10 , err= 1.6823973491942553e-11
ii= 948 , relres= 1.0913050080686977e-10 , err= 1.6803022466009852e-11
ii= 949 , relres= 1.1050704131974755e-10 , err= 1.6782854808064106e-11
ii= 950 , relres= 1.1850580795304661e-10 , err= 1.676368837126332e-11
ii= 951 , relres= 1.1872877312733804e-10 , err= 1.6745715326135672e-11
ii= 952 , relres= 1.2649572983746315e-10 , err= 1.672902370264619e-11
ii= 953 , relres= 1.2416274657394605e-10 , err= 1.6713611726263702e-11
ii= 954 , relres= 1.1282613410106595e-10 , err= 1.669937844723482e-11
ii= 955 , relres= 1.2155328742570623e-10 , err= 1.668612785738495e-11
ii= 956 , relres= 1.1818655994466396e-10 , err= 1.667360872346991e-11
ii= 957 , relres= 1.230435330814872e-10 , err= 1.6661547886740236e-11
ii= 958 , relres= 1.1777587055292114e-10 , err= 1.6649647056890163e-11
ii= 959 , relres= 1.3623496346207346e-10 , err= 1.663763689808475e-11
ii= 960 , relres= 1.308698581139826e-10 , err= 1.6625255129821886e-11
ii= 961 , relres= 1.251900530664417e-10 , err= 1.661229374646388e-11
ii= 962 , relres= 1.356405636221119e-10 , err= 1.6598603961765025e-11
ii= 963 , relres= 1.6292104005640935e-10 , err= 1.658414253013568e-11
ii= 964 , relres= 1.722711809564668e-10 , err= 1.6569022163682572e-11
ii= 965 , relres= 1.5037507786600712e-10 , err= 1.6553519727076898e-11
ii= 966 , relres= 1.3779772109006686e-10 , err= 1.6538065617764227e-11
ii= 967 , relres= 1.446376354590776e-10 , err= 1.6523165835775788e-11
ii= 968 , relres= 1.4767995283834817e-10 , err= 1.6509293013651183e-11
ii= 969 , relres= 1.314567371190105e-10 , err= 1.6496767005930464e-11
ii= 970 , relres= 1.531675479552863e-10 , err= 1.6485660407161476e-11
ii= 971 , relres= 1.3589754927991517e-10 , err= 1.6475812568107794e-11
ii= 972 , relres= 1.65065552975192142e-10 , err= 1.6466907570599423e-11
ii= 973 , relres= 1.2015181734963904e-10 , err= 1.645855036644461e-11
ii= 974 , relres= 1.4523939384638037e-10 , err= 1.6450365634340154e-11
ii= 975 , relres= 1.351405731509259e-10 , err= 1.6441978249008094e-11
ii= 976 , relres= 1.3739160116355934e-10 , err= 1.6433090066255305e-11
ii= 977 , relres= 1.3442258789619657e-10 , err= 1.6423434071842043e-11

```

```

ii= 978 , relres= 1.3018614468619888e-10 , err= 1.6412796914434977e-11
ii= 979 , relres= 1.2276019383287136e-10 , err= 1.640100952807859e-11
ii= 980 , relres= 1.4105190641481455e-10 , err= 1.6387983092046074e-11
ii= 981 , relres= 1.6501740712319676e-10 , err= 1.6373739576570514e-11
ii= 982 , relres= 1.175874207898251e-10 , err= 1.6358427613796535e-11
ii= 983 , relres= 1.5114586197010377e-10 , err= 1.634233165953434e-11
ii= 984 , relres= 1.1630257315641554e-10 , err= 1.6325837590144978e-11
ii= 985 , relres= 1.2010366031901895e-10 , err= 1.630938780037573e-11
ii= 986 , relres= 1.1529483043769726e-10 , err= 1.6293394771941346e-11
ii= 987 , relres= 1.115257489544179e-10 , err= 1.627818187452028e-11
ii= 988 , relres= 1.4220665384415265e-10 , err= 1.6263956676259474e-11
ii= 989 , relres= 1.1276298979517393e-10 , err= 1.6250801673246413e-11
ii= 990 , relres= 1.0822725490017017e-10 , err= 1.6238717027180928e-11
ii= 991 , relres= 1.0595918819048878e-10 , err= 1.6227627308631583e-11
ii= 992 , relres= 1.1536035280748892e-10 , err= 1.6217440509411043e-11
ii= 993 , relres= 1.034472760021121e-10 , err= 1.6208068427566454e-11
ii= 994 , relres= 1.2988053810315972e-10 , err= 1.6199132071256443e-11
ii= 995 , relres= 1.1478886756956033e-10 , err= 1.6190595642569925e-11
ii= 996 , relres= 1.1000421771751079e-10 , err= 1.6182441168107224e-11
ii= 997 , relres= 1.1341912317076773e-10 , err= 1.617465262909975e-11
ii= 998 , relres= 1.3044027069787686e-10 , err= 1.6167208347309644e-11
ii= 999 , relres= 1.1927266776994921e-10 , err= 1.6160069104627452e-11
FULL: none
ii= 0 , relres= 1.6429433272590805 , err= 0.9348462528972493
ii= 1 , relres= 1.4100333109153322 , err= 0.8878647518090131
ii= 2 , relres= 1.2131483362137276 , err= 0.8424382522391447
ii= 3 , relres= 1.1365671207955745 , err= 0.8050632398730682
ii= 4 , relres= 1.0558991264873916 , err= 0.7608521403352624
ii= 5 , relres= 0.9667932050555316 , err= 0.7267392297673525
ii= 6 , relres= 0.8519320013404825 , err= 0.676940050683272
ii= 7 , relres= 0.8056774866216733 , err= 0.6283641674945492
ii= 8 , relres= 0.7565760383350171 , err= 0.5866038566363992
ii= 9 , relres= 0.7377475028970887 , err= 0.5480271490497894
ii= 10 , relres= 0.5960159058098968 , err= 0.5119932800905044
ii= 11 , relres= 0.5650793357061442 , err= 0.4748893025420545
ii= 12 , relres= 0.5340230567348974 , err= 0.4403364094142731
ii= 13 , relres= 0.4810209781968884 , err= 0.4096449809132506
ii= 14 , relres= 0.4556473088672123 , err= 0.37881955366850745
ii= 15 , relres= 0.4077194401803662 , err= 0.3459731502662687
ii= 16 , relres= 0.3885062599905125 , err= 0.3149991664844079
ii= 17 , relres= 0.33517352235835257 , err= 0.2873687977389128
ii= 18 , relres= 0.336514524049781 , err= 0.26201675059566243
ii= 19 , relres= 0.2965787041152387 , err= 0.23984642721385788
ii= 20 , relres= 0.286489099536866 , err= 0.2190017852968143
ii= 21 , relres= 0.2560401222359263 , err= 0.20347923100667695
ii= 22 , relres= 0.21657980214348452 , err= 0.18830170823665207
ii= 23 , relres= 0.19683155481470518 , err= 0.17499814030725325
ii= 24 , relres= 0.17118564814695858 , err= 0.1633662965811086
ii= 25 , relres= 0.1560453905149484 , err= 0.15209802534007796
ii= 26 , relres= 0.15171516502759022 , err= 0.14185916393432094
ii= 27 , relres= 0.13091394713377788 , err= 0.13267073549004424
ii= 28 , relres= 0.14869367394699304 , err= 0.12573021436541845
ii= 29 , relres= 0.11837287528626454 , err= 0.1200207810344997
ii= 30 , relres= 0.1082753657222804 , err= 0.11333078572966672
ii= 31 , relres= 0.10074630586518403 , err= 0.1056021436250928
ii= 32 , relres= 0.08995953817846754 , err= 0.0995056632159608
ii= 33 , relres= 0.07863186456931065 , err= 0.09331777941323709
ii= 34 , relres= 0.07456927899737811 , err= 0.08755115516855624
ii= 35 , relres= 0.06982861232059359 , err= 0.08283910549269133
ii= 36 , relres= 0.06353785921189918 , err= 0.07788826080532602
ii= 37 , relres= 0.0610063288249896 , err= 0.07264909729219672

```

```
ii= 38 , relres= 0.05510658203365597 , err= 0.06895938910394132
ii= 39 , relres= 0.05694593689889055 , err= 0.06536417889796449
ii= 40 , relres= 0.051361018481644094 , err= 0.062469199623464707
ii= 41 , relres= 0.044590383407225265 , err= 0.05881131794361812
ii= 42 , relres= 0.043160722451239565 , err= 0.05565732337443235
ii= 43 , relres= 0.04188111044417477 , err= 0.05205435368509781
ii= 44 , relres= 0.03718109868510161 , err= 0.04865977485489514
ii= 45 , relres= 0.0412357760459102 , err= 0.04585917035317035
ii= 46 , relres= 0.03534195681579236 , err= 0.04339550513803978
ii= 47 , relres= 0.029155966016823808 , err= 0.040152905186275616
ii= 48 , relres= 0.028200399423729768 , err= 0.03719725383676715
ii= 49 , relres= 0.02781030219615545 , err= 0.03460426096178819
ii= 50 , relres= 0.02669939902991789 , err= 0.032239358468776916
ii= 51 , relres= 0.025960511687795115 , err= 0.030107173692206664
ii= 52 , relres= 0.028685986616009733 , err= 0.028450740992088098
ii= 53 , relres= 0.02150526358348649 , err= 0.026649824538232062
ii= 54 , relres= 0.021291276851682556 , err= 0.024458657462205366
ii= 55 , relres= 0.0201945742950709 , err= 0.022492116862183324
ii= 56 , relres= 0.0171102829142679 , err= 0.020388831280660907
ii= 57 , relres= 0.017084400739373 , err= 0.018624596204355637
ii= 58 , relres= 0.01621138224011635 , err= 0.017284061403815946
ii= 59 , relres= 0.014806268629972488 , err= 0.015663073387797266
ii= 60 , relres= 0.016074879628623417 , err= 0.014356599368769485
ii= 61 , relres= 0.012882772234094293 , err= 0.013244666067104307
ii= 62 , relres= 0.011873894137257732 , err= 0.012139202609603706
ii= 63 , relres= 0.011045335870993761 , err= 0.010942663410505258
ii= 64 , relres= 0.01000681599890385 , err= 0.010021757864663866
ii= 65 , relres= 0.009229465909831753 , err= 0.009124056958290055
ii= 66 , relres= 0.009405409054624003 , err= 0.008413180433046962
ii= 67 , relres= 0.008661117572810392 , err= 0.007808213147672865
ii= 68 , relres= 0.006690744823323098 , err= 0.007121152718838395
ii= 69 , relres= 0.006001932071439497 , err= 0.0064727884584021315
ii= 70 , relres= 0.005747411869971589 , err= 0.00587024181761741
ii= 71 , relres= 0.006371336150475133 , err= 0.005444992982044978
ii= 72 , relres= 0.0051991803348283 , err= 0.005157903569968919
ii= 73 , relres= 0.005232442691272971 , err= 0.004703279605635793
ii= 74 , relres= 0.005106237897209223 , err= 0.004440767447378513
ii= 75 , relres= 0.0044689923236324 , err= 0.004187670310722821
ii= 76 , relres= 0.0036929496254998776 , err= 0.0038583805246244743
ii= 77 , relres= 0.0031969839691873606 , err= 0.0035202321811423614
ii= 78 , relres= 0.0035548583982205123 , err= 0.0032620240601961576
ii= 79 , relres= 0.003183456086768151 , err= 0.003087857604091221
ii= 80 , relres= 0.002609723878588745 , err= 0.0028850645229553094
ii= 81 , relres= 0.002378668980731375 , err= 0.0026434004565093635
ii= 82 , relres= 0.0020720637830356392 , err= 0.002439859481496961
ii= 83 , relres= 0.0018666645633176012 , err= 0.0022374902540355512
ii= 84 , relres= 0.0016981274194775955 , err= 0.0020519885103762866
ii= 85 , relres= 0.0018860638418650032 , err= 0.0019067569883259724
ii= 86 , relres= 0.0015564788462126185 , err= 0.0017947662121523052
ii= 87 , relres= 0.0015023874315966725 , err= 0.0016647267462589564
ii= 88 , relres= 0.001269060787042625 , err= 0.0015440514951129407
ii= 89 , relres= 0.001101402464422839 , err= 0.0014275792748491805
ii= 90 , relres= 0.0010440020761393234 , err= 0.0013036797890868437
ii= 91 , relres= 0.0009735192633093789 , err= 0.0011950086773112917
ii= 92 , relres= 0.0011531597622749308 , err= 0.0011202585265567297
ii= 93 , relres= 0.0009222363973226848 , err= 0.001050860440199708
ii= 94 , relres= 0.0010059413133271051 , err= 0.0009730060188100272
ii= 95 , relres= 0.0008885701009861222 , err= 0.0009284655283862898
ii= 96 , relres= 0.000744295738069943 , err= 0.0008569966588408101
ii= 97 , relres= 0.0007519212000593662 , err= 0.0008056962190354038
ii= 98 , relres= 0.0006143486124224951 , err= 0.0007486051694960802
```

```
ii= 99 , relres= 0.0005941062273855886 , err= 0.000675886117337449
ii= 100 , relres= 0.0006579670880940318 , err= 0.0006187871075568828
ii= 101 , relres= 0.0005847835025797593 , err= 0.0005823473862787428
ii= 102 , relres= 0.0005253445533340712 , err= 0.0005296989275468368
ii= 103 , relres= 0.0005430308393695488 , err= 0.00049199552143495
ii= 104 , relres= 0.00042558951376192356 , err= 0.0004527604638408332
ii= 105 , relres= 0.0004610031951117865 , err= 0.0004169559420747334
ii= 106 , relres= 0.00036147562064520015 , err= 0.00038709857252219924
ii= 107 , relres= 0.000332432569514193 , err= 0.0003482880042902313
ii= 108 , relres= 0.0003537792775269952 , err= 0.0003216501214461728
ii= 109 , relres= 0.0003126931508491671 , err= 0.0002967389073093838
ii= 110 , relres= 0.00031279539546727263 , err= 0.0002726165421901171
ii= 111 , relres= 0.0002792056208865484 , err= 0.00025496847767208635
ii= 112 , relres= 0.00020385519940008495 , err= 0.0002300661525235331
ii= 113 , relres= 0.00019239958854262292 , err= 0.0002053041766678831
ii= 114 , relres= 0.0002065660561877998 , err= 0.00018487882731299199
ii= 115 , relres= 0.00017898035759737825 , err= 0.00017235089708305418
ii= 116 , relres= 0.00017533601357749956 , err= 0.00015707113573949324
ii= 117 , relres= 0.00018742402512388775 , err= 0.00014873242558419898
ii= 118 , relres= 0.0001372707889706463 , err= 0.00013599038976634588
ii= 119 , relres= 0.00014832100117429282 , err= 0.0001238708602280857
ii= 120 , relres= 0.00011741620115183929 , err= 0.00011369456701975306
ii= 121 , relres= 0.00010873114297360634 , err= 0.00010083504080687895
ii= 122 , relres= 0.000107699638422656 , err= 9.038028017877001e-05
ii= 123 , relres= 9.444119123242666e-05 , err= 8.327129852550612e-05
ii= 124 , relres= 7.644885530198327e-05 , err= 7.538507331874334e-05
ii= 125 , relres= 8.484899487056672e-05 , err= 6.820623968814758e-05
ii= 126 , relres= 6.808606830772349e-05 , err= 6.301736592559675e-05
ii= 127 , relres= 5.88178429544871e-05 , err= 5.6895317935816424e-05
ii= 128 , relres= 5.3177370002807075e-05 , err= 5.156669576496698e-05
ii= 129 , relres= 6.0205756327436224e-05 , err= 4.814766582492175e-05
ii= 130 , relres= 5.303484113485674e-05 , err= 4.4550625456311816e-05
ii= 131 , relres= 4.7203677096019317e-05 , err= 4.177068367489795e-05
ii= 132 , relres= 3.8073816663672525e-05 , err= 3.797378373296411e-05
ii= 133 , relres= 4.291154807288145e-05 , err= 3.480412196349747e-05
ii= 134 , relres= 3.338260145013546e-05 , err= 3.22590422597835e-05
ii= 135 , relres= 3.397542702145663e-05 , err= 2.9174847894139247e-05
ii= 136 , relres= 3.150095697616242e-05 , err= 2.757767630866265e-05
ii= 137 , relres= 3.2108227910415615e-05 , err= 2.5054127998793328e-05
ii= 138 , relres= 2.8533245907762018e-05 , err= 2.371502356333995e-05
ii= 139 , relres= 2.486822931399408e-05 , err= 2.130846015800878e-05
ii= 140 , relres= 2.253450009246764e-05 , err= 1.9534076118174524e-05
ii= 141 , relres= 2.0303632172551017e-05 , err= 1.7600808197691996e-05
ii= 142 , relres= 2.0534210733983405e-05 , err= 1.6239641186999266e-05
ii= 143 , relres= 1.8669820484461863e-05 , err= 1.4834086569622275e-05
ii= 144 , relres= 1.7385591934457464e-05 , err= 1.3814928191528258e-05
ii= 145 , relres= 1.3467359614472386e-05 , err= 1.2610723131875177e-05
ii= 146 , relres= 1.4039673113398486e-05 , err= 1.1401401321773153e-05
ii= 147 , relres= 1.2408344724171365e-05 , err= 1.0392793006164053e-05
ii= 148 , relres= 1.1927421888202492e-05 , err= 9.436029422963976e-06
ii= 149 , relres= 1.1397222361786844e-05 , err= 8.754370312197298e-06
ii= 150 , relres= 8.456992919218325e-06 , err= 7.928316397249159e-06
ii= 151 , relres= 8.687015470492692e-06 , err= 6.996258380958908e-06
ii= 152 , relres= 7.772408703404183e-06 , err= 6.3854846744308435e-06
ii= 153 , relres= 6.527971813881e-06 , err= 5.808520181821794e-06
ii= 154 , relres= 5.975332998269278e-06 , err= 5.138992888538799e-06
ii= 155 , relres= 5.900946915662839e-06 , err= 4.632150508862059e-06
ii= 156 , relres= 5.075708025481977e-06 , err= 4.298047770118791e-06
ii= 157 , relres= 5.392847414293911e-06 , err= 3.938516333444478e-06
ii= 158 , relres= 4.5513861795538944e-06 , err= 3.7355026763580324e-06
ii= 159 , relres= 4.694771825510447e-06 , err= 3.4622394374237387e-06
```

```

ii= 160 , relres= 3.526439888108695e-06 , err= 3.187539785967967e-06
ii= 161 , relres= 2.9995192244958917e-06 , err= 2.901949468656515e-06
ii= 162 , relres= 3.542658901636626e-06 , err= 2.632360225774344e-06
ii= 163 , relres= 3.139939451274149e-06 , err= 2.4588559926562073e-06
ii= 164 , relres= 2.651866307633406e-06 , err= 2.287294397322683e-06
ii= 165 , relres= 2.1324218828072297e-06 , err= 2.0724746213557477e-06
ii= 166 , relres= 1.8066909128045914e-06 , err= 1.855661669688862e-06
ii= 167 , relres= 1.7081637207826948e-06 , err= 1.666525364249137e-06
ii= 168 , relres= 1.5267758935670355e-06 , err= 1.5019189762845613e-06
ii= 169 , relres= 1.5278135029594218e-06 , err= 1.3740133995801505e-06
ii= 170 , relres= 1.3411432733342552e-06 , err= 1.280924234364094e-06
ii= 171 , relres= 1.4042751688227608e-06 , err= 1.205785678767236e-06
ii= 172 , relres= 1.1912643580298632e-06 , err= 1.1133403932775633e-06
ii= 173 , relres= 1.2110487515981322e-06 , err= 1.0399563113678965e-06
ii= 174 , relres= 1.0519621891027686e-06 , err= 9.61758882107861e-07
ii= 175 , relres= 1.0036944439729242e-06 , err= 9.041385560331879e-07
ii= 176 , relres= 9.285992333244983e-07 , err= 8.315491066281811e-07
ii= 177 , relres= 8.141945094857408e-07 , err= 7.846469827703502e-07
ii= 178 , relres= 7.191184503285195e-07 , err= 7.179251200276176e-07
ii= 179 , relres= 7.922780232333845e-07 , err= 6.886450791295478e-07
ii= 180 , relres= 6.171990255166896e-07 , err= 6.450297144667232e-07
ii= 181 , relres= 5.277559024609902e-07 , err= 5.98809998449235e-07
ii= 182 , relres= 4.832190803815705e-07 , err= 5.507263447483079e-07
ii= 183 , relres= 4.1027095336029287e-07 , err= 5.04028725599348e-07
ii= 184 , relres= 4.906874901121289e-07 , err= 4.7407364689570627e-07
ii= 185 , relres= 4.1022111903034973e-07 , err= 4.466860639218873e-07
ii= 186 , relres= 3.500911514728032e-07 , err= 4.0969902374071716e-07
ii= 187 , relres= 3.6262823521758997e-07 , err= 3.790284433731905e-07
ii= 188 , relres= 3.2212127411216494e-07 , err= 3.563507236023936e-07
ii= 189 , relres= 2.8340839954505616e-07 , err= 3.231063664236581e-07
ii= 190 , relres= 2.9574436515386965e-07 , err= 2.998264833622243e-07
ii= 191 , relres= 2.8912598626421027e-07 , err= 2.768344520004472e-07
ii= 192 , relres= 2.572140592801704e-07 , err= 2.614780914010283e-07
ii= 193 , relres= 2.54931188671122e-07 , err= 2.416034998927791e-07
ii= 194 , relres= 2.0867323145919667e-07 , err= 2.255117808386943e-07
ii= 195 , relres= 1.7104994966511474e-07 , err= 2.048473269351878e-07
ii= 196 , relres= 1.791221624117025e-07 , err= 1.8517780951087846e-07
ii= 197 , relres= 1.716121821026858e-07 , err= 1.749725396658687e-07
ii= 198 , relres= 1.6171647719631452e-07 , err= 1.6015371255344554e-07
ii= 199 , relres= 1.5933455861518735e-07 , err= 1.536311703349418e-07
ii= 200 , relres= 1.247864731464859e-07 , err= 1.419580301002334e-07
ii= 201 , relres= 1.280409056641574e-07 , err= 1.3072506597536476e-07
ii= 202 , relres= 9.72726599621826e-08 , err= 1.205501002629495e-07
ii= 203 , relres= 1.1533408022030187e-07 , err= 1.1158143763679084e-07
ii= 204 , relres= 8.848697121306797e-08 , err= 1.0337016027182143e-07
ii= 205 , relres= 8.299176916852229e-08 , err= 9.505038975409234e-08
ii= 206 , relres= 8.835373403672602e-08 , err= 8.889778316483278e-08
ii= 207 , relres= 8.444262318370918e-08 , err= 8.402151775883001e-08
ii= 208 , relres= 6.194044711408252e-08 , err= 7.802318351349525e-08
ii= 209 , relres= 6.479419024805184e-08 , err= 7.164722948601679e-08
ii= 210 , relres= 6.345817193565565e-08 , err= 6.764815375263671e-08
ii= 211 , relres= 4.8479020050649056e-08 , err= 6.179668980136917e-08
ii= 212 , relres= 6.32587851226389e-08 , err= 5.75211308475059e-08
ii= 213 , relres= 4.735433746612617e-08 , err= 5.3567894315773585e-08
ii= 214 , relres= 4.8089676982224506e-08 , err= 4.9557008841691426e-08
ii= 215 , relres= 3.951678118223922e-08 , err= 4.5458621107461394e-08
ii= 216 , relres= 4.2181498783244205e-08 , err= 4.214543950825733e-08
ii= 217 , relres= 3.3140713987417544e-08 , err= 3.9418980252708313e-08
ii= 218 , relres= 3.823050059176946e-08 , err= 3.5975477436421675e-08
ii= 219 , relres= 3.331208829330704e-08 , err= 3.455950191414896e-08
ii= 220 , relres= 2.5065761144809412e-08 , err= 3.170275586346439e-08

```

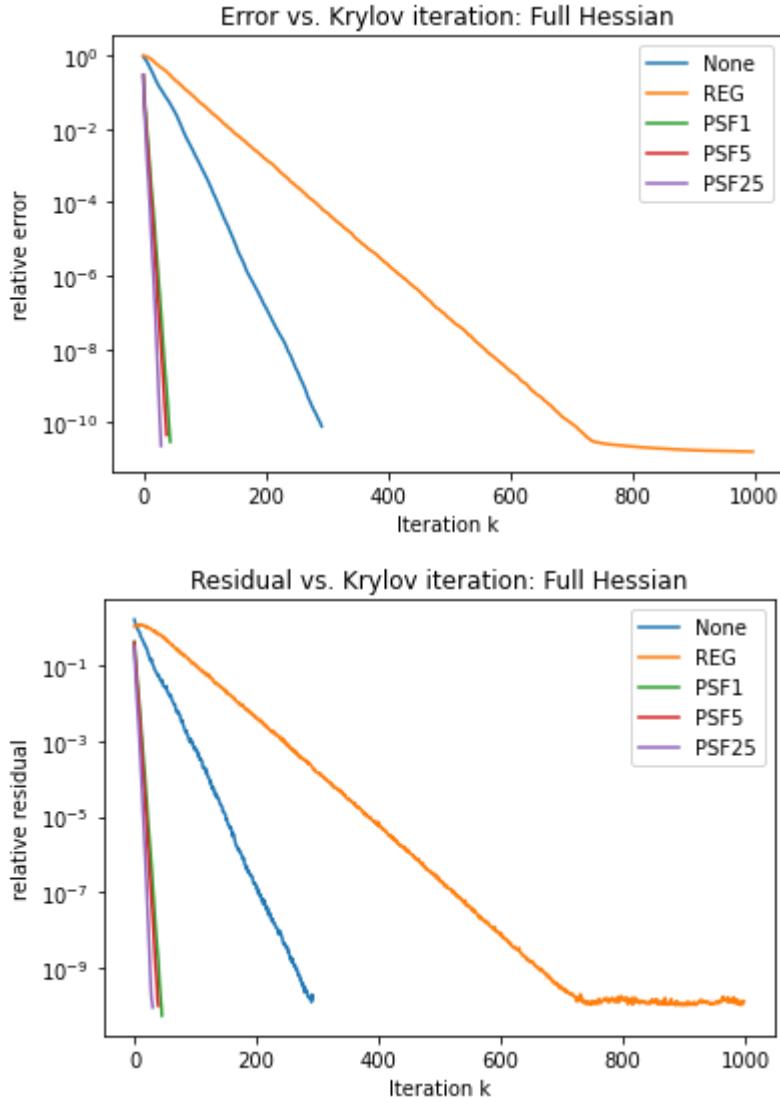
```
ii= 221 , relres= 2.8860241932771477e-08 , err= 2.9799962035973787e-08
ii= 222 , relres= 2.5454834924602053e-08 , err= 2.7984083445565152e-08
ii= 223 , relres= 2.552065425674811e-08 , err= 2.6606999117223657e-08
ii= 224 , relres= 1.911767694062388e-08 , err= 2.4829696667520497e-08
ii= 225 , relres= 1.7423269836815235e-08 , err= 2.2755076347205914e-08
ii= 226 , relres= 1.970342828643933e-08 , err= 2.1438147766778078e-08
ii= 227 , relres= 1.6430276401479152e-08 , err= 2.027591414516609e-08
ii= 228 , relres= 1.5499856675314607e-08 , err= 1.8923405841931063e-08
ii= 229 , relres= 1.57171171533103e-08 , err= 1.7648622436866583e-08
ii= 230 , relres= 1.3695866252448915e-08 , err= 1.638941407748006e-08
ii= 231 , relres= 1.2416873652707762e-08 , err= 1.5113243112284344e-08
ii= 232 , relres= 1.2215430666036173e-08 , err= 1.399202549005852e-08
ii= 233 , relres= 1.0786826498995947e-08 , err= 1.3063628510432322e-08
ii= 234 , relres= 1.0683996731048281e-08 , err= 1.1817137493888987e-08
ii= 235 , relres= 1.0189884937436934e-08 , err= 1.1114776077699507e-08
ii= 236 , relres= 7.678552297599952e-09 , err= 1.0003614939848737e-08
ii= 237 , relres= 8.029707414980157e-09 , err= 8.853526550638049e-09
ii= 238 , relres= 9.574593198008477e-09 , err= 8.307937711656157e-09
ii= 239 , relres= 7.905625777390487e-09 , err= 7.690438383440639e-09
ii= 240 , relres= 7.127376797082418e-09 , err= 7.199091083384902e-09
ii= 241 , relres= 7.192694289952272e-09 , err= 6.804168624588963e-09
ii= 242 , relres= 6.074746969287017e-09 , err= 6.172882916091396e-09
ii= 243 , relres= 5.332626201745887e-09 , err= 5.734985465972503e-09
ii= 244 , relres= 4.314408816239411e-09 , err= 5.173182759851026e-09
ii= 245 , relres= 4.270859397009214e-09 , err= 4.560295510200819e-09
ii= 246 , relres= 4.446194110796108e-09 , err= 4.215911661093336e-09
ii= 247 , relres= 3.7372334067424465e-09 , err= 3.823066469168338e-09
ii= 248 , relres= 3.909514391506024e-09 , err= 3.448990533661054e-09
ii= 249 , relres= 3.5719793068418926e-09 , err= 3.2127572775479077e-09
ii= 250 , relres= 3.0631374730040118e-09 , err= 2.98855659677076e-09
ii= 251 , relres= 3.3172407055935726e-09 , err= 2.7054593865361074e-09
ii= 252 , relres= 2.5511082250195506e-09 , err= 2.5558881855833783e-09
ii= 253 , relres= 2.569581162917567e-09 , err= 2.32783036160018e-09
ii= 254 , relres= 2.136967667292469e-09 , err= 2.207473368479409e-09
ii= 255 , relres= 1.7118192545410174e-09 , err= 1.9589833751151887e-09
ii= 256 , relres= 1.9190343133708686e-09 , err= 1.7834185689429172e-09
ii= 257 , relres= 1.5403691305231909e-09 , err= 1.6310696007032682e-09
ii= 258 , relres= 1.3892955176929458e-09 , err= 1.4773208128888945e-09
ii= 259 , relres= 1.728776157896325e-09 , err= 1.3730074670823947e-09
ii= 260 , relres= 1.4422495130162696e-09 , err= 1.2944673389362285e-09
ii= 261 , relres= 1.1687523485243574e-09 , err= 1.187115532291714e-09
ii= 262 , relres= 1.148820427209828e-09 , err= 1.0793671515377508e-09
ii= 263 , relres= 1.0272036170554078e-09 , err= 9.893436552718412e-10
ii= 264 , relres= 1.0930051405983496e-09 , err= 8.945268103961947e-10
ii= 265 , relres= 8.731998982410794e-10 , err= 8.342652725814054e-10
ii= 266 , relres= 8.660699436066377e-10 , err= 7.288937953862631e-10
ii= 267 , relres= 8.635316556263847e-10 , err= 6.750234101436073e-10
ii= 268 , relres= 6.260006464242308e-10 , err= 5.868329392895824e-10
ii= 269 , relres= 6.572530359848781e-10 , err= 5.180508734223282e-10
ii= 270 , relres= 6.371061696918299e-10 , err= 4.815502401574311e-10
ii= 271 , relres= 5.966008377372671e-10 , err= 4.4303201344829313e-10
ii= 272 , relres= 4.737952040899482e-10 , err= 4.0531406116053493e-10
ii= 273 , relres= 4.872343280675839e-10 , err= 3.762659143928167e-10
ii= 274 , relres= 3.787250783375293e-10 , err= 3.3977811092823323e-10
ii= 275 , relres= 3.946107266417856e-10 , err= 3.1464022843356937e-10
ii= 276 , relres= 3.105778630488768e-10 , err= 2.8253480470755356e-10
ii= 277 , relres= 3.1980677709773527e-10 , err= 2.6758796508322594e-10
ii= 278 , relres= 2.3057594736048147e-10 , err= 2.432338322033216e-10
ii= 279 , relres= 3.2223969381086666e-10 , err= 2.224104449433559e-10
ii= 280 , relres= 2.2896463371612136e-10 , err= 2.1127062852861242e-10
ii= 281 , relres= 2.5860702772023264e-10 , err= 1.9655060202957588e-10
```

```

ii= 282 , relres= 2.1862559001546582e-10 , err= 1.8312575609311833e-10
ii= 283 , relres= 2.2011082591499168e-10 , err= 1.7292463908766678e-10
ii= 284 , relres= 1.700921336637584e-10 , err= 1.5589260224643024e-10
ii= 285 , relres= 1.6787728906152026e-10 , err= 1.4073710427251453e-10
ii= 286 , relres= 1.706739331497458e-10 , err= 1.2949187535182803e-10
ii= 287 , relres= 1.6057755364406e-10 , err= 1.2150588824653483e-10
ii= 288 , relres= 1.6166915821804573e-10 , err= 1.1270581233479493e-10
ii= 289 , relres= 1.4759357390195644e-10 , err= 1.0529255366669509e-10
ii= 290 , relres= 1.2337222446789014e-10 , err= 9.700769513237978e-11
ii= 291 , relres= 1.9507594279388244e-10 , err= 8.796422021518565e-11
ii= 292 , relres= 1.518438897153181e-10 , err= 7.905827767038302e-11
ii= 293 , relres= 9.953864550876054e-11 , err= 7.341541724773748e-11
<matplotlib.legend.Legend at 0x7f3abf274e80>

```

Out[51]:



Plots to make (5):

- CG convergence (Full Newton, full page width, residual)

In [52]:

```

np.savetxt(save_dir_str + '/cg_relres_none.txt', all_relres_none)
np.savetxt(save_dir_str + '/cg_relres_reg.txt', all_relres_reg)
for nb, relres in zip(all_num_batches, all_relres):
    np.savetxt(save_dir_str + '/cg_relres_p' + str(nb) + '.txt', relres)

```

Preconditioned spectrum plots

In [53]:

```
all_H_pch_dense = list()
for nb, H in zip(all_num_batches, all_H_pch):
    print('building p' + str(nb) + '_dense')
    all_H_pch_dense.append(build_dense_matrix_from_matvecs(H.matvec, Vh2.dim()))
```

building p1_dense
 building p5_dense
 building p25_dense

In [54]:

```
print('building reg_dense')
reg_dense = build_dense_matrix_from_matvecs(REG.apply_R_numpy, Vh2.dim())
```

building reg_dense

In [55]:

```
print('building H_dense')
H_dense = build_dense_matrix_from_matvecs(H_linop.matvec, Vh2.dim())

print('building Hgn_dense')
Hgn_dense = build_dense_matrix_from_matvecs(Hgn_linop.matvec, Vh2.dim())
```

building H_dense
 building Hgn_dense

In [56]:

```
all_ee_gn = list()
all_P_gn = list()
for nb, Hpch_dense in zip(all_num_batches, all_H_pch_dense):
    print('computing ee' + str(nb) + '_gn')
    ee_gn, P_gn = sla.eigh(Hgn_dense, Hpch_dense)
    all_ee_gn.append(ee_gn)
    all_P_gn.append(P_gn)

print('computing eeR_gn')
eeR_gn, PR_gn = sla.eigh(Hgn_dense, reg_dense)
print('computing ee_none_gn')
ee_none_gn, P_none_gn = sla.eigh(Hgn_dense)

all_cond_gn = list()
for ee_gn in all_ee_gn:
    all_cond_gn.append(np.max(ee_gn) / np.min(ee_gn))
condR_gn = np.max(eeR_gn) / np.min(eeR_gn)
cond_none_gn = np.max(ee_none_gn) / np.min(ee_none_gn)

for nb, cond_gn in zip(all_num_batches, all_cond_gn):
    print('cond' + str(nb) + '_gn=' , cond_gn)

print('condR_gn=' , condR_gn)
print('cond_none_gn=' , cond_none_gn)
```

computing ee1_gn

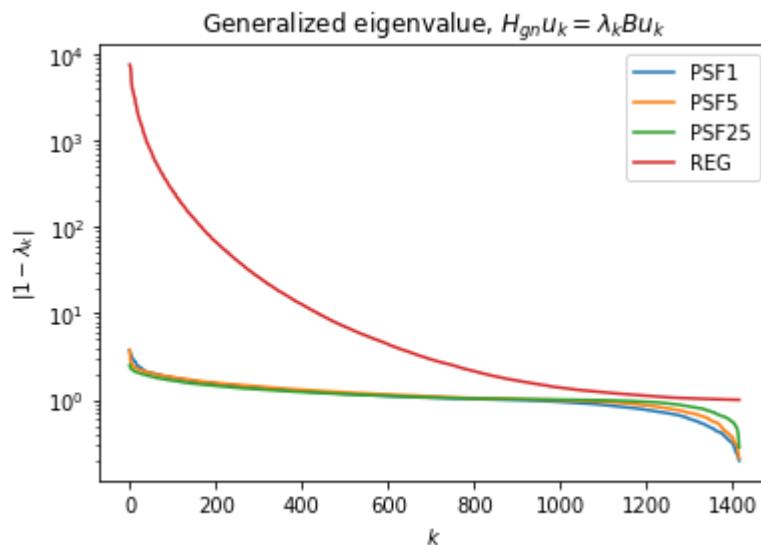
```
computing ee5_gn
computing ee25_gn
computing eeR_gn
computing ee_none_gn
cond1_gn= 19.09179376713973
cond5_gn= 17.80228689923624
cond25_gn= 8.932119380135207
condR_gn= 7640.964640412505
cond_none_gn= 972.1505567784867
```

In [57]:

```
plt.figure()
for ee_gn in all_ee_gn:
#    plt.semilogy(np.sort(np.abs(1.0 - ee_gn))[:-1])
    plt.semilogy(np.sort(ee_gn)[:-1])

# plt.semilogy(np.sort(np.abs(1.0 - eeR_gn))[:-1])
plt.semilogy(np.sort(eeR_gn)[:-1])
plt.legend(['PSF' + str(nb) for nb in all_num_batches] + ['REG', 'None'])
plt.xlabel(r'$k$')
plt.ylabel(r'$|1 - \lambda_k|$')
plt.title(r'Generalized eigenvalue, $H_{gn} u_k = \lambda_k B u_k$')
```

Out[57]:



In [58]:

```
all_ee = list()
all_P = list()
for nb, Hpch_dense in zip(all_num_batches, all_H_pch_dense):
    print('computing ee' + str(nb))
    ee, P = sla.eigh(H_dense, Hpch_dense)
    all_ee.append(ee)
    all_P.append(P)

print('computing eeR')
eeR, PR = sla.eigh(H_dense, reg_dense)
print('computing ee_none')
ee_none, P_none = sla.eigh(H_dense)

all_cond = list()
for ee in all_ee:
    all_cond.append(np.max(ee) / np.min(ee))
condR = np.max(eeR) / np.min(eeR)
```

```

cond_none = np.max(ee_none) / np.min(ee_none)

for nb, cond in zip(all_num_batches, all_cond):
    print('cond' + str(nb) + '=' , cond)

print('condR=' , condR)
print('cond_none=' , cond_none)

```

```

computing ee1
computing ee5
computing ee25
computing eeR
computing ee_none
cond1= 19.00142342245143
cond5= 17.795941453773036
cond25= 8.889097756657046
condR= 7640.986530049078
cond_none= 917.6858215976631

```

In [59]:

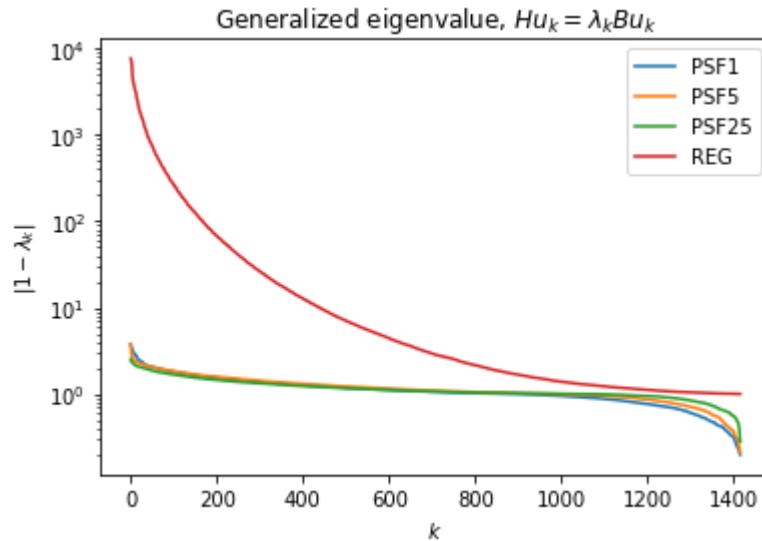
```

plt.figure()
for ee in all_ee:
#     plt.semilogy(np.sort(np.abs(1.0 - ee))[:-1])
    plt.semilogy(np.sort(ee)[::-1])

# plt.semilogy(np.sort(np.abs(1.0 - eeR))[:-1])
plt.semilogy(np.sort(eeR)[::-1])
plt.legend(['PSF' + str(nb) for nb in all_num_batches] + ['REG', 'None'])
plt.xlabel(r'$k$')
plt.ylabel(r'$|1 - \lambda_k|$')
plt.title(r'Generalized eigenvalue, $H u_k = \lambda_k B u_k$')

```

Out[59]: Text(0.5, 1.0, 'Generalized eigenvalue, \$H u_k = \lambda_k B u_k\$')



Plots to make (6)

- Preconditioned spectrum (Full newton)
- Condition number table (Gauss-Newton and full Newton)

In [61]:

```

for nb, ee in zip(all_num_batches, all_ee):
    np.savetxt(save_dir_str + '/generalized_eigenvalues_psf' + str(nb) + '.txt',
    np.savetxt(save_dir_str + '/generalized_eigenvalues_reg.txt', eeR)

for nb, ee_gn in zip(all_num_batches, all_ee_gn):
    np.savetxt(save_dir_str + '/generalized_eigenvalues_psf' + str(nb) + '_gn.txt',
    np.savetxt(save_dir_str + '/generalized_eigenvalues_reg_gn.txt', eeR_gn)

np.savetxt(save_dir_str + '/all_num_batches.txt', all_num_batches)
np.savetxt(save_dir_str + '/cond_psf.txt', all_cond)
np.savetxt(save_dir_str + '/condR.txt', np.array([condR]))
np.savetxt(save_dir_str + '/cond_none.txt', np.array([cond_none]))

np.savetxt(save_dir_str + '/cond_psf_gn.txt', all_cond_gn)
np.savetxt(save_dir_str + '/condR_gn.txt', np.array([condR_gn]))
np.savetxt(save_dir_str + '/cond_none_gn.txt', np.array([cond_none_gn]))

ps = " *14 + " | " + "{:>8}" .format("None") + " " + "{:>8}" .format("REG") + " "
for nb in all_num_batches:
    ps += "{:>8}" .format("PSF" + str(nb)) + " "
ps += "\n"
ps += "-"*(34 + 9*len(all_num_batches)) + "\n"
ps += "cond(P^-1 H) | "
ps += "{:>8}" .format(np.format_float_scientific(cond_none, precision=2, trim='k'))
ps += "{:>8}" .format(np.format_float_scientific(condR, precision=2, trim='k', uppercase=True))
for c in all_cond:
    ps += "{:>8}" .format(np.format_float_scientific(c, precision=2, trim='k', uppercase=True))
ps += "\n"
ps += "cond(P^-1 Hgn) | "
ps += "{:>8}" .format(np.format_float_scientific(cond_none_gn, precision=2, trim='k'))
ps += "{:>8}" .format(np.format_float_scientific(condR_gn, precision=2, trim='k', uppercase=True))
for c_gn in all_cond_gn:
    ps += "{:>8}" .format(np.format_float_scientific(c_gn, precision=2, trim='k', uppercase=True))
ps += "\n"

print(ps)

#     ps += "{:>9}" .format(np.format_float_scientific(g_psf, precision=2, trim='k'))
#     ps += " | "
#     ps += "{:<3d}" .format(ncg_reg)

#-----| None   Reg   PSF3   PSF6   PSF9
#-----| 923.5  7762  15.59  14.16  13.96
# P^-1 H | 978.0  7760  15.57  14.12  13.98

```

	None	Reg	PSF3	PSF6	PSF9
# P^-1 H	923.5	7762	15.59	14.16	13.96
# P^-1 Hgn	978.0	7760	15.57	14.12	13.98

	None	REG	PSF1	PSF5	PSF25
cond(P^-1 H)	9.18e+02	7.64e+03	1.90e+01	1.78e+01	8.89e+00
cond(P^-1 Hgn)	9.72e+02	7.64e+03	1.91e+01	1.78e+01	8.93e+00

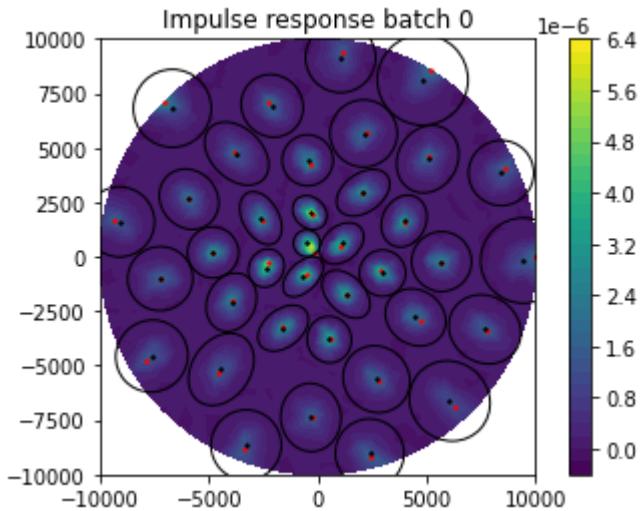
Plots to make (7):

- Batches of impulse responses (Nick will do this, save as hi-res PNG)

In [62]:

```
PCH1.PCK.col_batches.visualize_impulse_response_batch(0)
```

```
/home/nick/repos/nalger_helper_functions/nalger_helper_functions/plot_ellipse.p
y:16: RuntimeWarning: divide by zero encountered in double_scalars
    theta = np.arctan(v_big[1] / v_big[0]) * 180. / np.pi
```



Regularization robustness plot

```
In [ ]: # all_gammas = np.logspace(6, 3, 7)

# b_G = np.random.randn(Vh2.dim())

# for gamma in all_gammas:
#     print('--- GAMMA ---:', gamma)
#     REG.update_gamma(gamma)

#     print('Building Regularization H-Matrix')
#     R_hmatrix_G = REG.make_R_hmatrix(Hd_pch_nonsym.bct, rtol=1e-6)

#     H_pch3_G = Hd_pch3 + R_hmatrix_G
#     H_pch6_G = Hd_pch6 + R_hmatrix_G
#     H_pch9_G = Hd_pch9 + R_hmatrix_G

#     preconditioner3_G = H_pch3_G.factorized_inverse()
#     preconditioner6_G = H_pch6_G.factorized_inverse()
#     preconditioner9_G = H_pch9_G.factorized_inverse()

#     p3_linop_G = spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=precondit
#     p6_linop_G = spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=precondit
#     p9_linop_G = spla.LinearOperator((Vh2.dim(), Vh2.dim()), matvec=precondit

#     print('GN: get truth:')
#     x_true_G, _, _, _ = cg_wrapper(Hgn_linop, b, p9_linop_G)

#     print('GN: p9')
#     _, _, all_relres_p9_G, all_err_p9_G = cg_wrapper(Hgn_linop, b_G, p9_linop_
#     print('GN: p6')
#     _, _, all_relres_p6_G, all_err_p6_G = cg_wrapper(Hgn_linop, b_G, p6_linop_
#     print('GN: p3')
#     _, _, all_relres_p3_G, all_err_p3_G = cg_wrapper(Hgn_linop, b_G, p3_linop_
#     print('GN: reg')
```

```
# _, _, all_relres_reg_G, all_err_reg_G = cg_wrapper(Hgn_linop, b_G, ireg_li
#     print('GN: none')
# _, _, all_relres_none_G, all_err_none_G = cg_wrapper(Hgn_linop, b_G, None,
```