Assignment 3
MDM18B028

Levenberg-Marquardt and Conjugate Gradient Method

CODE DOCUMENTATION

P.M.Kavyassree

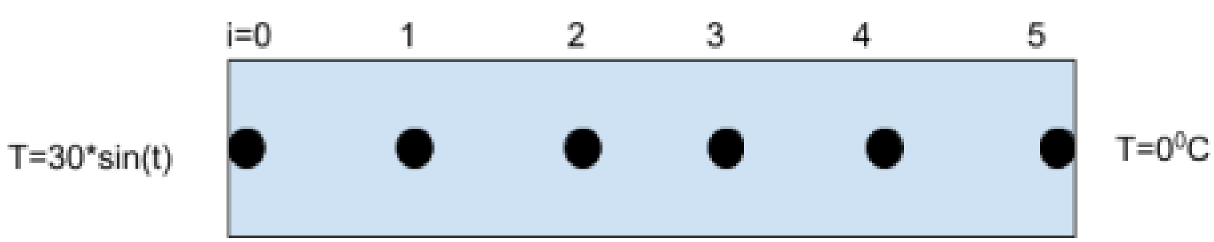
Problem Statement:

A steel rod is subjected to a variant time varying temperature -30 sin(t) on the left end and 0℃ on the right end.A sensor is placed at 0.03 m from the left end and temperature for each time step is noted for the forward model. For the inverse problem the temperature readings are used to estimate the parameter which is the scalar multiple of the time variant left boundary condition .The parameters are compared and the graphs are plotted.

Governing Equation

$(\partial T/\partial t) * (1/\alpha) = \partial^2 T/\partial x^2$

$\alpha = k/(\rho * c)$



Boundary conditions

$T_o^j=30*\sin(t)$

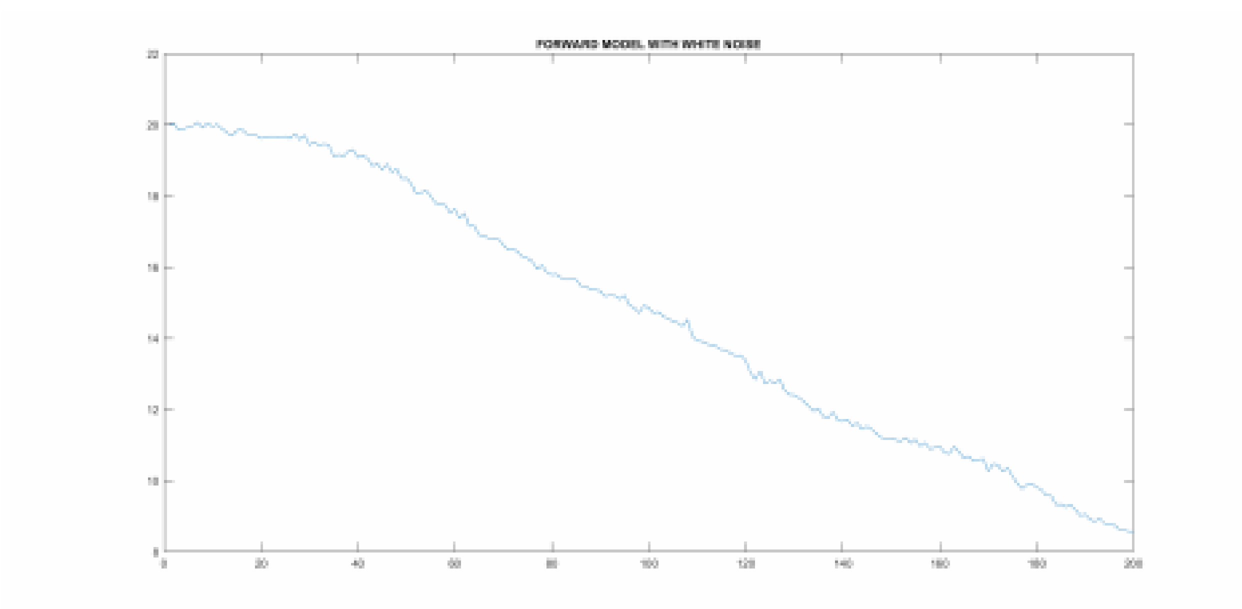
$T_5^j=0$℃ for all j=0,1,2,3

Parameters to be measured: The scalar multiple of the left boundary

Unknown parameters: Temperature variant at left boundary

Sensitivity coefficient: It's a 200*3(for 3 parameters) and 200 X 2 (for 2 parameters) .In the matlab code the sensitivity matrix is mentioned as **J.**The values inside the sensitivity matrix are the sensitivity coefficients.C

Forward model with white noise:



Levenberg-Marquardt Method:

The LMA is used in many software applications for solving generic curve-fitting problems. However, as with many fitting algorithms, the LMA finds only a local minimum, which is not necessarily the global minimum. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach.

The algorithm was first published in 1944 by Kenneth Levenberg, while working at the Frankford Army Arsenal. It was rediscovered in 1963 by Donald Marquardt, who worked as a statistician at DuPont, and independently by Girard, Wynne and Morrison.

This is one of the parameter estimation methods. For any inverse problem the forward model has to be curated and then proceed with the inverse problem. We will be using only a single sensor and using the LM method we will be predicting the parameter of the function used as the time varying heat conduction problem.

Algorithm:

Suppose that temperature measurements $\mathbf{Y}=(Y_1,Y_2,...,Y_I)$ are given at times t_i, $i=1,...,I$. Also, suppose an initial guess $\mathbf{P}^0$ is available for the vector of unknown parameters **P**. Choose a value for μ^0, say, $\mu^0 = 0.001$ and set $k=0$ [6]. Then,

Step 1. Solve the direct heat transfer problem given by equations (2.1.1) with the available estimate $\mathbf{P}^k$ in order to obtain the temperature vector $\mathbf{T}(\mathbf{P}^k)=(T_1,T_2,...,T_I)$.

Step 2. Compute $S(\mathbf{P}^k)$ from equation (2.1.3.b).

Step 3. Compute the sensitivity matrix $\mathbf{J}^k$ defined by equation (2.1.7.a) and then the matrix Ω^k given by equation (2.1.15), by using the current values of $\mathbf{P}^k$.

Step 4. Solve the following linear system of algebraic equations, obtained from the iterative procedure of the Levenberg-Marquardt Method, equation (2.1.13):

$$[(\mathbf{J}^k)^T\mathbf{J}^k+\mu^k\Omega^k]\Delta\mathbf{P}^k=(\mathbf{J}^k)^T[\mathbf{Y}-\mathbf{T}(\mathbf{P}^k)] \tag{2.1.16}$$

in order to compute $\Delta\mathbf{P}^k=\mathbf{P}^{k+1}-\mathbf{P}^k$

Step 5. Compute the new estimate $\mathbf{P}^{k+1}$ as

$$\mathbf{P}^{k+1}=\mathbf{P}^k+\Delta\mathbf{P}^k \tag{2.1.17}$$

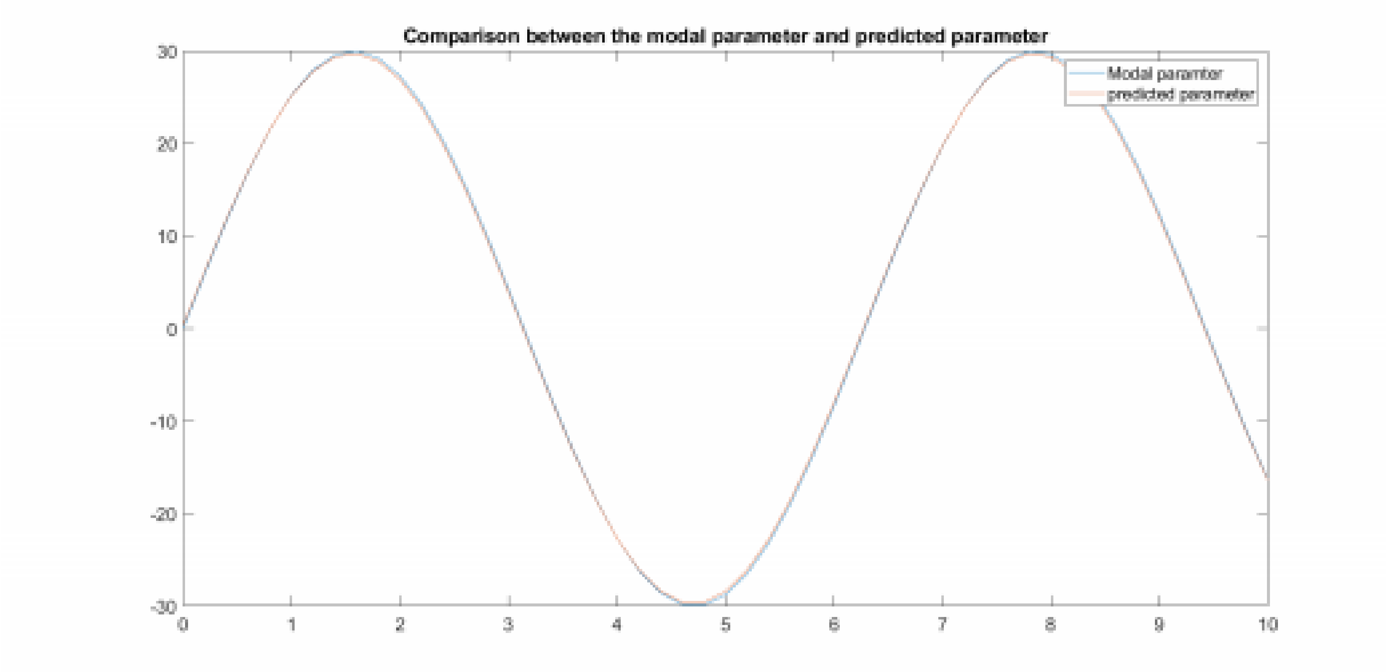
Step 6. Solve the direct problem (2.1.1) with the new estimate $\mathbf{P}^{k+1}$ in order to find $\mathbf{T}(\mathbf{P}^{k+1})$. Then compute $S(\mathbf{P}^{k+1})$, as defined by equation (2.1.3.b).

Step 7. If $S(\mathbf{P}^{k+1})\geq S(\mathbf{P}^k)$, replace μ^k by $10\mu^k$ and return to step 4.

Step 8. If $S(\mathbf{P}^{k+1})< S(\mathbf{P}^k)$, accept the new estimate $\mathbf{P}^{k+1}$ and replace μ^k by $0.1\mu^k$.

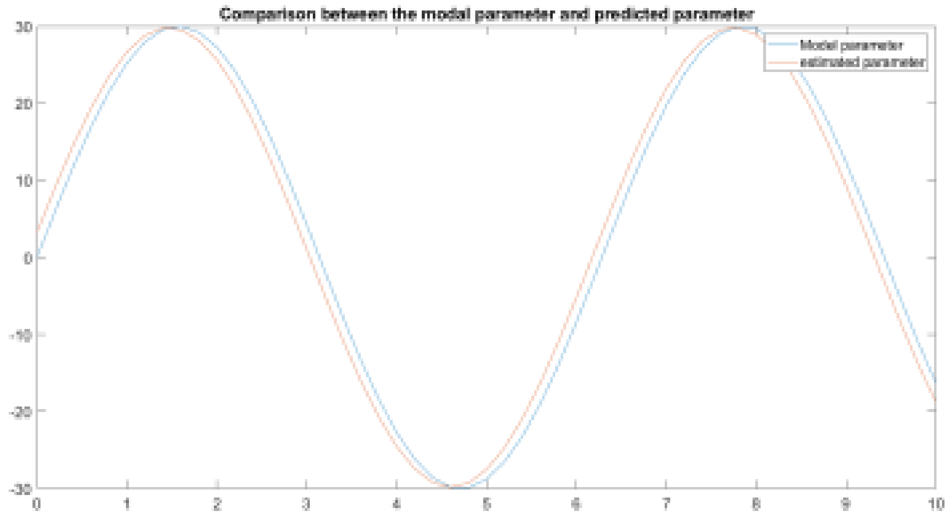
Step 9. Check the stopping criteria given by equations (2.1.14.a-c). Stop the iterative procedure if any of them is satisfied; otherwise, replace k by $k+1$ and return to step 3.

Two parameter results:



X=30*sin(t) -(model function)
Y=0.3316*cos(t)+29.7170*sin(t) -(predicted function)

Three parameter results:



X=30*sin(t)-(model function)
Y=3.2019*cos(t)+29.6569*sin(t)+0.0002*t^2 -(predicted function)

Conjugate gradient Method:

In mathematics, the conjugate gradient method is an algorithm for the numerical solution of particular systems of linear equations, namely those whose matrix is positive-definite. The conjugate gradient method is often implemented as an iterative algorithm, applicable to sparse systems that are too large to be handled by a direct implementation or other direct methods such as the Cholesky decomposition. Large sparse systems often arise when numerically solving partial differential equations or optimization problems.

The conjugate gradient method can also be used to solve unconstrained optimization problems such as energy minimization. It is commonly attributed to Magnus Hestenes and Eduard Stiefel, who programmed it on the Z4, and extensively researched.

The biconjugate gradient method provides a generalization to non-symmetric matrices. Various nonlinear conjugate gradient methods seek minima of nonlinear equations and black-box objective functions.

Algorithm:

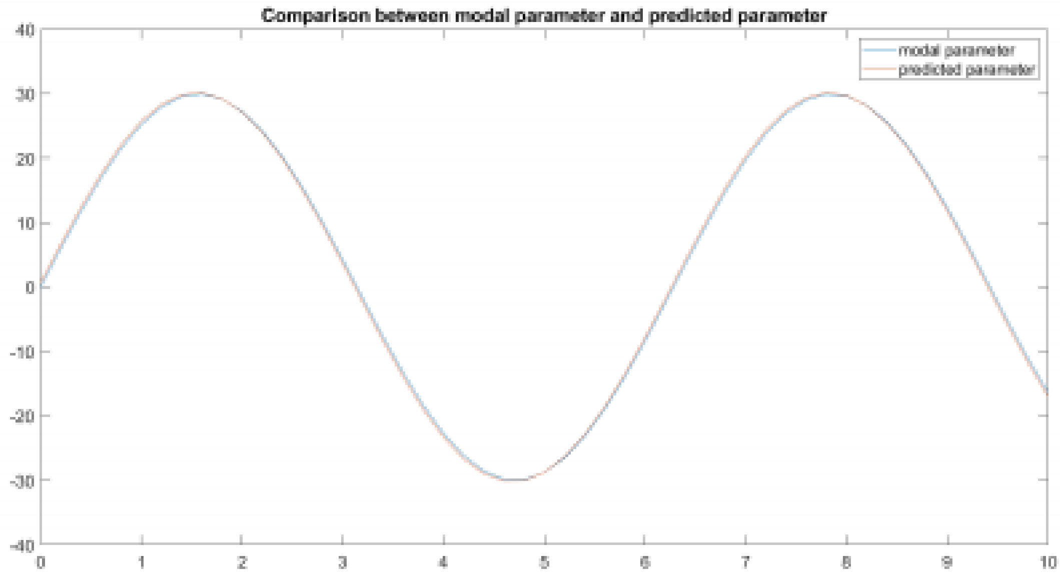
Suppose that temperature measurements $\mathbf{Y}=(Y_1,Y_2,...,Y_I)$ are given at times t_i, i = 1, ..., I, and an initial guess $\mathbf{P}^0$ is available for the vector of unknown parameters $\mathbf{P}$. Set k = 0 and then

Step 1. Solve the direct heat transfer problem (2.1.1) by using the available estimate $\mathbf{P}^k$ and obtain the vector of estimated temperatures $\mathbf{T}(\mathbf{P}^k)=(T_1,T_2,...,T_I)$.
Step 2. Check the stopping criterion given by equation (2.2.8). Continue if not satisfied.
Step 3. Compute the sensitivity matrix $\mathbf{J}^k$ defined by equation (2.1.7.a), by using one of the appropriate methods described in section 2-1.
Step 4. Knowing $\mathbf{J}^k$, $\mathbf{Y}$ and $\mathbf{T}(\mathbf{P}^k)$, compute the gradient direction $\nabla S(\mathbf{P}^k)$ from equation (2.2.5.a) and then the conjugation coefficient γ^k from either equation (2.2.4.a) or (2.2.4.b).
Step 5. Compute the direction of descent $\mathbf{d}^k$ by using equation (2.2.3).
Step 6. Knowing $\mathbf{J}^k$, $\mathbf{Y}$, $\mathbf{T}(\mathbf{P}^k)$ and $\mathbf{d}^k$, compute the search step size β^k from equation (2.2.7.c).
Step 7. Knowing $\mathbf{P}^k$, β^k and $\mathbf{d}^k$, compute the new estimate $\mathbf{P}^{k+1}$ using equation (2.2.2).
Step 8. Replace k by k+1 and return to step 1.

Two parameter result:

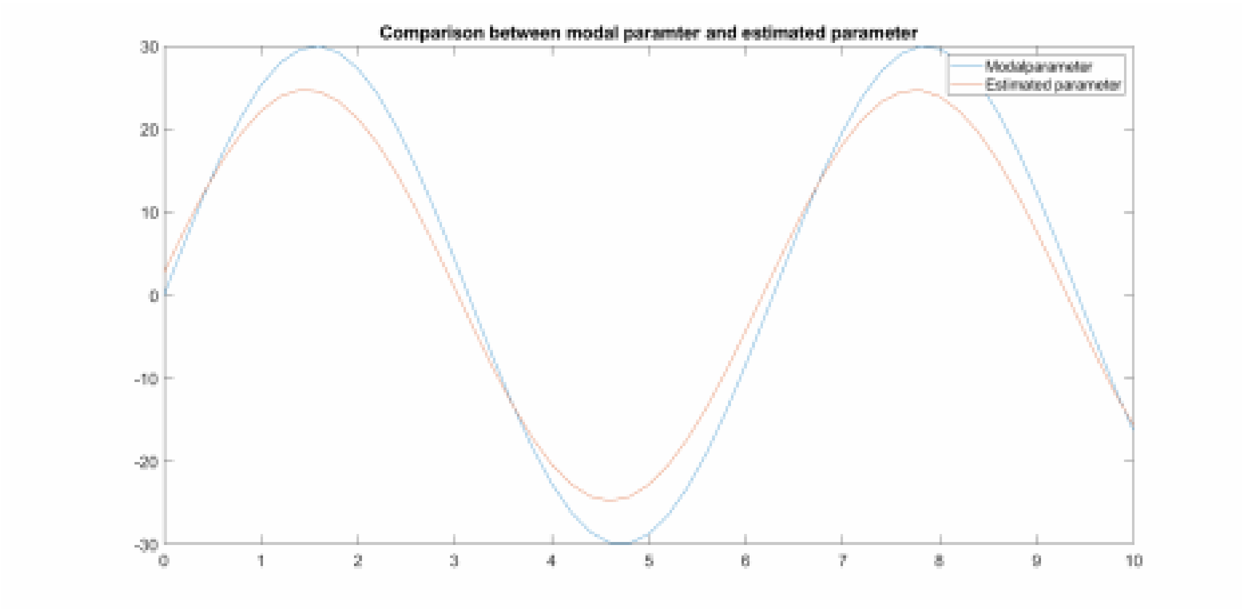
X=30*sin(t)-(modal function)

Y=0.6738*cos(t)+30.1842*sin(t)-(predicted function)



9

Three parameter result:



X=30*sin(t)-(modal function)
Y=2.6948*cos(t)+24.6121*sin(t)+-0.000585291775126205*t^2 -(predicted function)