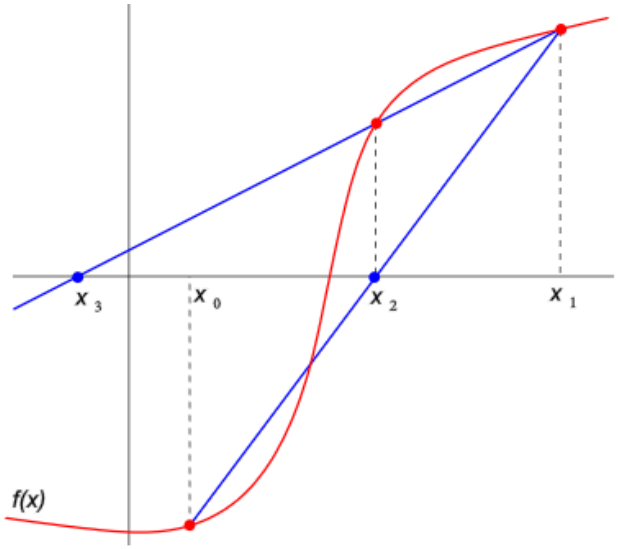
Secant Method – C Program

AUG 29, 2017 Manas Sharma (https://www.bragitoff.com/author/admin/)

Secant Method, is a Numerical Technique to find the root of an algebraic or transcendental equation.
The root is approximated by drawing secant lines repeatedly.



(https://www.bragitoff.com/wp-content/uploads/2017/08/Secant_method.png)
A secant line is a line joining two points on a function. Secant method requires two initial guesses(x0 and x1), to draw the first secant line. The root of this line(x2), that is, where this line touches the x-axis, becomes the new point, and now

a secant line is drawn between the new point(x2) and one of the last points(x1).
This process is repeated until a root is found upto a certain tolerance.

The method is similar to Bisection Method, in that it requires two initial guesses, but still a lot different, as the guesses don't require to bracket(enclose) the root. Moreover, unlike Bisection Method, Secant Method may not always converge, so it might be a good idea to have a limit for the maximum iterations to be performed.

So, the program would ask the user to enter two initial guesses:x1 and x2.
Then, it will calculate the new point(x3) using the formula:

$$x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)}$$

PROGRAM (Simple Version):

```
/**********************************
*********SECANT METHOD*************
**********************************/
#include<stdio.h>
#include<math.h>
/*Function whose root is to be determined*/
double f(double x){
    return x*x-4;
}
main(){
    int iter=1,maxSteps;
    double x1,x2,x3,eps;
    printf("Enter the accuracy desired: \n");
    scanf("%lf",&eps);
    printf("Enter the intial guesses: \nx1 = ");
    scanf("%lf",&x1);
    printf("x2 = ");
    scanf("%lf",&x2);
    printf("Enter the max number of iterations to be performed: ");
    scanf("%d",&maxSteps);
    printf("iter\tx1\t\tx2\t\tx3\t\tf(x3)\n");
    printf("_____________________________________________________________________\n");
    do{
        x3=(x1*f(x2)-x2*f(x1))/(f(x2)-f(x1));
        printf("%d\t%lf\t%lf\t%lf\t%lf\n",iter,x1,x2,x3,f(x3));
        x1=x2;
        x2=x3;
        iter++;
    }while(fabs(f(x3))>eps&&iter<=maxSteps);
    printf("\nOne of the roots is: %lf",x3);
}
```

OUTPUT:

For x^3-27:

```
Enter the accuracy desired:
0.00001
Enter the intial guesses:
x1 = 5
x2 = 10
iter    x1              x2              x3              f(x3)
_____________________________________________________________________
1       5.000000        10.000000       4.440000        60.528384
2       10.000000       4.440000        4.071180        40.477798
3       4.440000        4.071180        3.326612        9.813446
4       4.071180        3.326612        3.088330        2.455808
5       3.326612        3.088330        3.008796        0.238203
6       3.088330        3.008796        3.000254        0.006845
7       3.008796        3.000254        3.000001        0.000020
8       3.000254        3.000001        3.000000        0.000000

One of the roots is: 3.000000
```

(https://www.bragitoff.com/wp-content/uploads/2017/08/secant-method-output-1-c-program.png)

For x*x-4:

```
Enter the accuracy desired:
0.001
Enter the intial guesses:
x1 = 0
x2 = 5
iter    x1              x2              x3              f(x3)
_____________________________________________________________
1       0.000000        5.000000        0.800000        -3.360000
2       5.000000        0.800000        1.379310        -2.097503
3       0.800000        1.379310        2.341772        1.483897
4       1.379310        2.341772        1.942991        -0.224785
5       2.341772        1.942991        1.995453        -0.018168
6       1.942991        1.995453        2.000066        0.000263

One of the roots is: 2.000066
-------------------------------
```

(https://www.bragitoff.com/wp-content/uploads/2017/08/secant-method-output-2-c-program.png)

PROGRAM (USING FUNCTIONS)

```c
/**********************************
*********SECANT METHOD**************
**********************************/
#include<stdio.h>
#include<math.h>
/*Function whose root is to be determined*/
double f(double x){
    return x*x-4;
}
/*Function that returns the root from Secant Method*/
double secant(double f(double x), double x1, double x2, double eps, int maxSteps){
    int iter=1;
    double x3;
    do{
        x3=(x1*f(x2)-x2*f(x1))/(f(x2)-f(x1));
        x1=x2;
        x2=x3;
        iter++;
    }while(fabs(f(x3))>eps&&iter<=maxSteps);
    return x3;
}
/*Secant Method Function that tabulates the values at each iteration*/
double secantPrint(double f(double x), double x1, double x2, double eps, int maxSteps){
    int iter=1;
    double x3;
    printf("_____________________________________________________________\n");
    printf("iter\tx1\t\tx2\t\tx3\t\tf(x3)\n");
    printf("_____________________________________________________________\n");
    do{
        x3=(x1*f(x2)-x2*f(x1))/(f(x2)-f(x1));
        printf("%d\t%lf\t%lf\t%lf\t%lf\n",iter,x1,x2,x3,f(x3));
        x1=x2;
        x2=x3;
        iter++;
    }while(fabs(f(x3))>eps&&iter<=maxSteps);
    printf("_____________________________________________________________\n");
    return x3;
}
main(){
    int maxSteps;
    double x1,x2,x3,eps;
    printf("Enter the accuracy desired: \n");
    scanf("%lf",&eps);
    printf("Enter the intial guesses: \nx1 = ");
    scanf("%lf",&x1);
    printf("x2 = ");
    scanf("%lf",&x2);
    printf("Enter the max number of iterations to be performed: ");
    scanf("%d",&maxSteps);
    printf("\nOne of the roots is: %lf",secantPrint(f,x1,x2,eps,maxSteps));
}
```

OUTPUT:

```
Enter the accuracy desired:
0.001
Enter the intial guesses:
x1 = 0
x2 = 5
Enter the max number of iterations to be performed: 15

iter    x1              x2              x3              f(x3)
---------------------------------------------------------------
1       0.000000        5.000000        0.800000        -3.360000
2       5.000000        0.800000        1.379310        -2.097503
3       0.800000        1.379310        2.341772        1.483897
4       1.379310        2.341772        1.942991        -0.224785
5       2.341772        1.942991        1.995453        -0.018168
6       1.942991        1.995453        2.000066        0.000263
---------------------------------------------------------------

One of the roots is: 2.000066
--------------------------------
```

(https://www.bragitoff.com/wp-content/uploads/2017/08/secant-method-output-better-1-c-program.png)

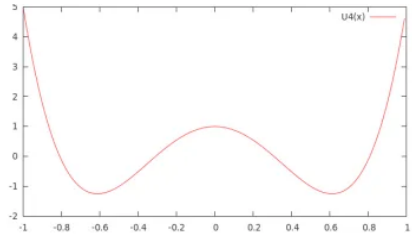
Manas Sharma (https://www.bragitoff.com/author/admin/)

PhD researcher at Friedrich-Schiller University Jena, Germany. I'm a physicist specializing in theoretical, computational and experimental condensed matter physics. I like to develop Physics related apps and softwares from time to time. Can code in most of the popular languages. Like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

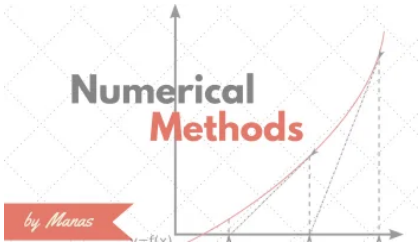
(https://www.facebook.com/ducktape07) (https://www.plus.google.com/u/0/+ManasSharma07)

(https://www.twitter.com/manassharma07) (https://www.youtube.com/manassharma100)

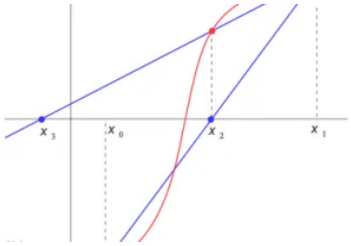
Related



(https://www.bragitoff.com/2017/08/numerical-root-finding-exercises-c-programs/)
Numerical Root Finding Exercise - C Programs (https://www.bragitoff.com/2017/08/numerical-root-finding-exercises-c-programs/)
August 29, 2017
In "C Programs"



(https://www.bragitoff.com/2017/07/numerical-methods-calculators-android-app-documentation/)
Numerical Methods Calculators- Android App [Documentation] (https://www.bragitoff.com/2017/07/numerical-methods-calculators-android-app-documentation/)
July 4, 2017
In "Android Studio"



(https://www.bragitoff.com/2017/09/numerical_root_finding_exercises_c_programming/)
Numerical Root Finding Exercises - C Programming (https://www.bragitoff.com/2017/09/numerical_root_finding_exercises_c_programming/)
September 1, 2017
In "C Programs"

[wpedon id="7041" align="center"]

Posted in C Programs (https://www.bragitoff.com/category/compu-geek/c-programs/), Numerical Analysis Programming (https://www.bragitoff.com/category/physics/numerical-analysis/), Root Finding (https://www.bragitoff.com/category/compu-geek/c-programs/root-finding/)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Notify me of follow-up comments by email.

Notify me of new posts by email.

FOLLOW US ON FB (HTTPS://WWW.FACEBOOK.COM/BRAGITOFF)

LATEST ARTICLES

Microstructure Classification using Neural Networks – Streamlit App (https://www.bragitoff.com/2022/02/microstructure-classification-using-neural-networks-streamlit-app/)
Feb 9, 2022
(https://www.bragitoff.com/2022/02/microstructure-classification-using-neural-networks-streamlit-app/)

Handwritten digit recognition using CrysX-NN neural network trained on MNIST_Plus (https://www.bragitoff.com/2022/01/handwritten-digit-recognition-using-crysx-nn-neural-network-trained-on-mnist_plus/)
Jan 18, 2022
(https://www.bragitoff.com/2022/01/handwritten-digit-recognition-using-crysx-nn-neural-network-trained-on-mnist_plus/)

Handwritten digit recognition using CrysX – Neural Network library and Streamlit Web App (https://www.bragitoff.com/2022/01/handwritten-digit-recognition-using-crysx-neural-network-library-and-streamlit-web-app/)
Jan 16, 2022

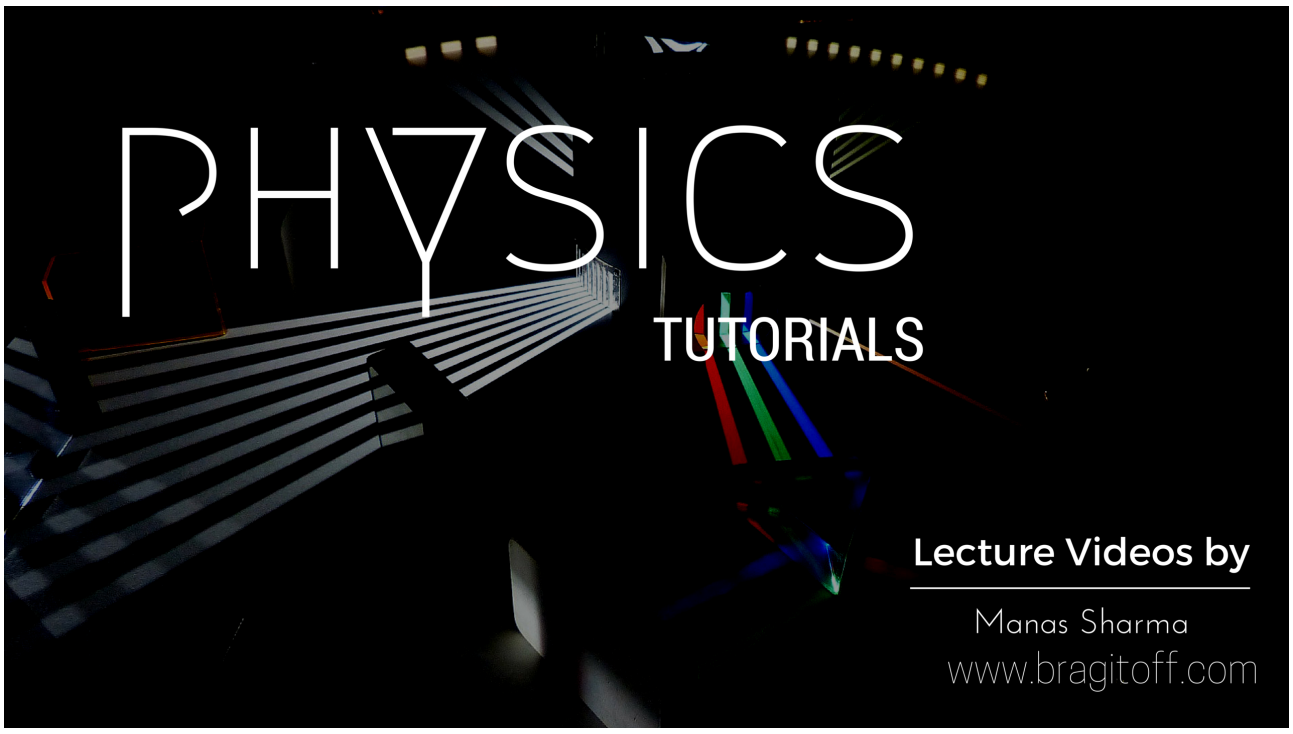
Digit recognition App (Streamlit) trained on the MNIST_Plus dataset using PyTorch CNN model (https://www.bragitoff.com/2022/01/digit-recognition-app-streamlit-trained-on-the-mnist_plus-dataset-using-pytorch-cnn-model/)
Jan 16, 2022

Streamlit App for Digit recognition trained on the MNIST dataset using PyTorch (https://www.bragitoff.com/2022/01/streamlit-app-for-digit-recognition-trained-on-the-mnist-dataset-using-pytorch/)
Jan 12, 2022

Mean Squared Error loss function and its gradient (derivative) for a batch of inputs [Python Code] (https://www.bragitoff.com/2021/12/mean-squared-error-loss-function-and-its-gradient-derivative-for-a-batch-of-inputs-python-code/)
Dec 30, 2021

(https://www.bragitoff.com/2021/12/mean-squared-error-loss-function-and-its-gradient-derivative-for-a-batch-of-inputs-python-code/)

SUBSCRIBE ON YOUTUBE



(https://www.youtube.com/channel/UCO0mdq1TnLctbwUGj_TnR7Q)

GET THE OFFICIAL BRAGITOFF APP

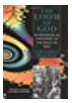
(https://play.google.com/store/apps/details?id=com.bragitoff.bragitoffbeta&utm_source=global_co&utm_medium=prtnr&utm_content=Mar2515&utm_campaign=PartBadge&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1)

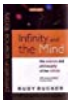
CURRENTLY READING

The Physics Book: From the Big Bang to Quantum Resurrection, 250 Milestones in the History of Physics (https://www.goodreads.com/review/show/1879038412?utm_medium=api&utm_source=custom_widget)
by Clifford A. Pickover (https://www.goodreads.com/author/show/381.Clifford_A_Pickover)

(https

Strange Beauty: Murray Gell-Mann and the Revolution in Twentieth-Century Physics (https://www.goodreads.com/review/show/1837071180?utm_medium=api&utm_source=custom_widget)
by George Johnson (https://www.goodreads.com/author/show/4158220.George_Johnson)


The Loom Of God (https://www.goodreads.com/review/show/1837069713?utm_medium=api&utm_source=custom_widget)
by Clifford A. Pickover (https://www.goodreads.com/author/show/381.Clifford_A_Pickover)


Infinity And The Mind (https://www.goodreads.com/review/show/1837069170?utm_medium=api&utm_source=custom_widget)
by Rudy Rucker (https://www.goodreads.com/author/show/130704.Rudy_Rucker)


goodreads (https://www.goodreads.com/)

ASK A QUESTION ON OUR FORUM

(http://physwhiz.bragitoff.com)

About (https://www.bragitoff.com/about/) Policies (https://www.bragitoff.com/policies/)

Apps (http://apps.bragitoff.com)

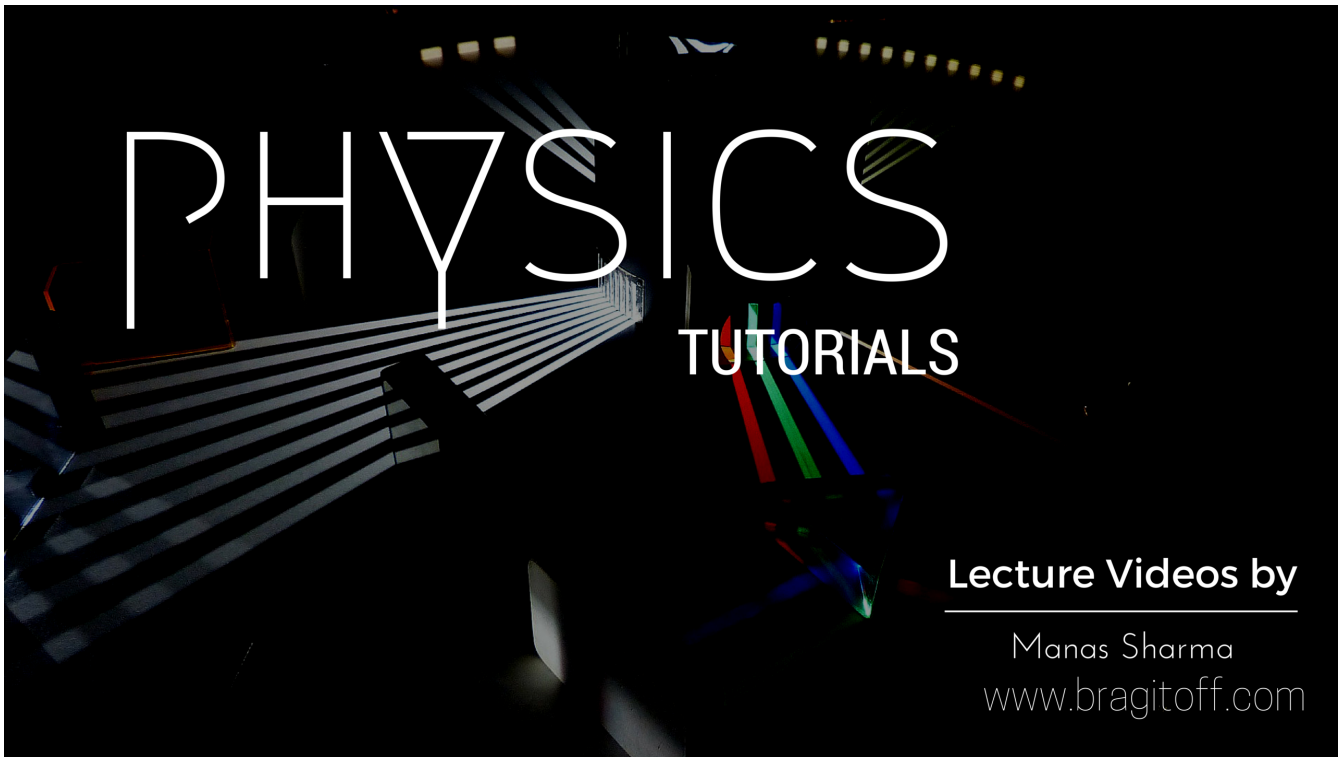
ALEXA RANK-HELP US : WRITE A REVIEW ABOUT US

(https://www.alexa.com/siteinfo/bragitoff.com)

SOCIAL

(https://www.facebook.com/ducktape07/) (https://twitter.com/manassharma07/)
(https://www.instagram.com/ducktape07/) (https://www.linkedin.com/in/manassharma24059a12a/)
(https://www.youtube.com/user/manassharma100/) (https://plus.google.com/u/0/+manassharma07/)
(https://profiles.wordpress.org/manassharma07/)

SUBSCRIBE ON YOUTUBE



(https://www.youtube.com/channel/UCO0mdq1TnLctbwUGj_TnR7Q)

GET THE OFFICIAL BRAGITOFF APP

(https://play.google.com/store/apps/details?id=com.bragitoff.bragitoffbeta&utm_source=global_co&utm_medium=prtnr&utm_content=Mar2515&utm_campaign=PartBadge&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1)