**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Ders : Sayısal Analiz**

**Dönem : 2020-2021 GÜZ Dönemi**

**Adı Soyadı: Muhammet Kemal Güvenç**

**Okul No: B181210076**

**Konu : Dönem Ödevi**

**Problemin Analizi**

Bir dairenin içini doldurmak için en az 3 tane üçgen gerekir. Eğer dairenin içindeki üçgen sayısına dersek olmalıdır. Dairenin içindeki n adet üçgenin alanlarının toplamına , dairenin alanına diyelim. arttıkça değeri değerine yaklaşır ve hata payı azalır.

Bu üçgenlerin tepe açılarına diyelim. Bu değer şu şekilde hesaplanır:

Dairenin içinde oluşan üçgenler ikizkenar üçgenlerdir. Eşit olan kenarları dairenin yarıçapına eşittir. Taban kenarlarına dersek kosinüs teoremi ile taban kenarları şu şekilde hesaplanır:

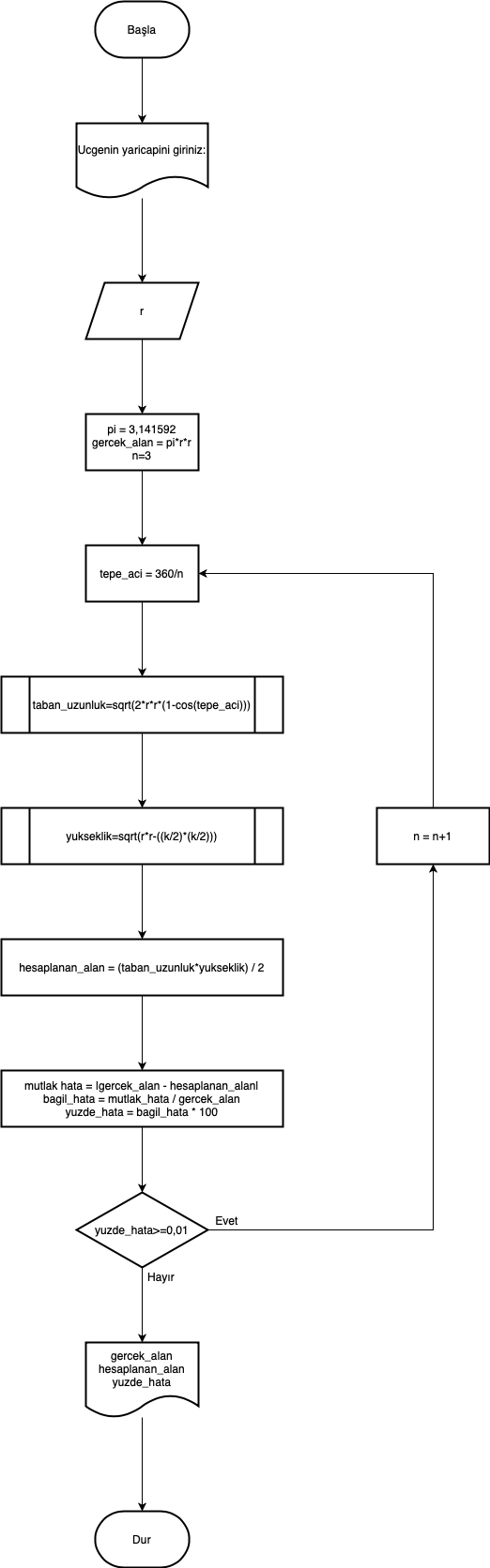
Yüksekliğe h derse yükseklik Pisagor bağıntısı ile şu şekilde hesaplanır:

Bir üçgenin alanı (taban uzunluğu x yükseklik) / 2 olduğunu biliyoruz. Buna dayanarak üçgenlerin alanı şöyle hesaplanır:

Eğer bütün üçgenlerin alanlarını toplarsak ’i buluruz. adet üçgen için şu şekildedir:

olduğunu biliyoruz. ’in ’ye göre hata yüzdesi şu şekilde hesaplanır:

**Akış Şeması**



**Algoritmanın Açıklaması**

Kullanıcıdan ilk önce dairenin yarıçapı isteniyor. Daha sonra bu bilgi kullanılarak problemin analizi kısmında yapılan hesaplamaların aynısını iteratif bir şekilde yaparak %0,01 hata yüzdesi ile dairenin alanını hesaplar. Akış diyagramında “sqrt” adında kök alan ve “cos” adında açının kosinüsünü alan 2 tane fonksiyon olduğu varsayılmıştır.

**Yazılım Kodu (C++)**

#include <iostream>

#include <cmath>

**using** **namespace** std;

**int** main(**int** argc, **const** **char** \* argv[]) {

**int** n=3;

**double** pi=3.141592;

**double** r;

**double** gercek\_alan;

**double** aci;

**double** taban\_uzunluk;

**double** yukseklik;

**double** hesaplanan\_alan=0;

cout << "Ucgenin yaricapini giriniz: ";

cin >> r;

gercek\_alan=pi\*r\*r;

**do**{

aci=2\*pi/n;

taban\_uzunluk=sqrt(2\*r\*r\*(1-cos(aci)));

yukseklik=sqrt(r\*r-((taban\_uzunluk/2)\*(taban\_uzunluk/2)));

hesaplanan\_alan=n\*taban\_uzunluk\*yukseklik/2;

n=n+1;

}**while**(abs(gercek\_alan-hesaplanan\_alan)/gercek\_alan\*100>0.01);

cout << "Gercek Alan: " << gercek\_alan << endl;

cout << "Hesaplanan Alan: " << hesaplanan\_alan << endl;

cout << "Yuzde Hata: " << abs(gercek\_alan-hesaplanan\_alan)/gercek\_alan\*100 << endl;

**return** 0;

}

**Program Çıktıları**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu