

INS 2021 SAYISAL ÇÖZÜMLEME



Karadeniz Teknik Üniversitesi
İnşaat Mühendisliği Bölümü

TRABZON, 2022

Sayısal yöntemler, matematik problemlerinin, aritmetik işlemlerle çözülebilmelerini sağlayan tekniklerdir. Çeşitli sayısal yöntemler olmasına rağmen hepsinin ortak özelliği çok sayıda zahmetli aritmetik işlem içermesidir.

Sayısal yöntemleri öğrenmemizi gerektiren sebepler:

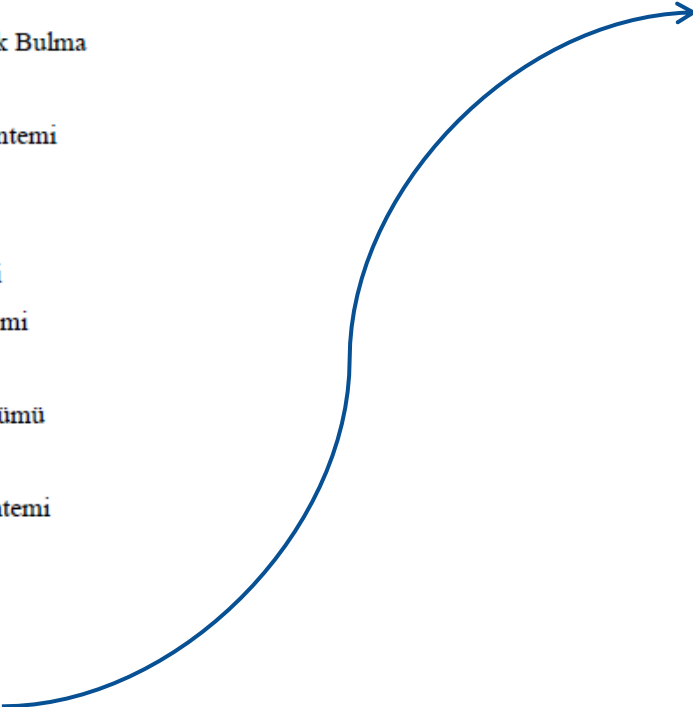
- 1) Sayısal yöntemler, yüksek matematik problemlerini basit aritmetik işlemler ile çözümlmesine olanak sağlamaktadır.
- 2) Mühendislik uygulamalarının çözümünde paket programların kullanımı günümüzde oldukça yaygındır. Bu programların işleyişinde sayısal yöntemler sıklıkla kullanılmaktadır. Bu nedenle sayısal yöntemlerin mantığının bilinmesi söz konusu programların etkili bir şekilde kullanılmasına faydası olacaktır.
- 3) Bilgisayar programlama ve sayısal yöntemler hakkında bilgisi olan bir kişinin kendi problemini çözebilecek bilgisayar programı yazması mümkündür.
- 4) Mühendislikte birçok doğrusal (lineer) olmayan diferansiyel denklem analitik olarak çözülememektedir. Fakat sayısal olarak çözülebilmektedir.

A hand holding a pen, writing on a blue background with binary code and a stylized building.

1. İÇİNDEKİLER

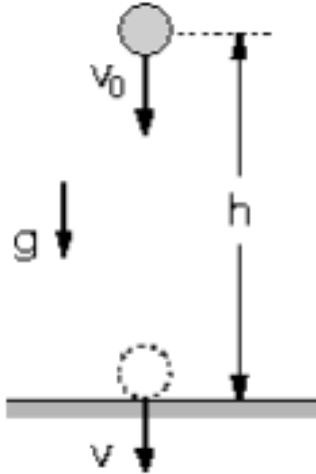
İÇİNDEKİLER

1. Giriş, Hata Analizi
2. Lineer Olmayan Denklemlerde Kök Bulma
 - 2.1. Kapalı Yöntemler
 - 2.1.1. Aralığı İkiye Bölme Yöntemi
 - 2.1.2. Regula-Falsi Yöntemi
 - 2.2. Açık Yöntemler
 - 2.2.1. Basit İterasyon Yöntemi
 - 2.2.2. Newton-Raphson Yöntemi
 - 2.2.3. Sekant Yöntemi
3. Lineer Denklem Sistemlerinin Çözümü
 - 3.1. Direkt Yöntemler
 - 3.1.1. Gauss-Eliminasyon Yöntemi
 - 3.1.2. Pivottlama
 - 3.1.3. Gauss-Jordan Yöntemi
 - 3.1.4. LU / LDU Yöntemleri
 - 3.1.5. Cholesky Yöntemi

- 
- 3.2. Bir Matrisin Tersi
 - 3.3. Ardışık Yaklaşım Yöntemleri
 - 3.3.1. Jacobi Yöntemi
 - 3.3.2. Gauss-Seidel Yöntemi
 4. Doğrusal Olmayan Denklem Sistemlerinin Çözümü
 - 4.1. Ardışık Yaklaşım Yöntemi
 - 4.2. Newton-Raphson Yöntemi
 5. Sayısal Türev
 6. Sayısal İntegral
 - 6.1. Trapez Kuralı
 - 6.2. Simpson Kuralı

GİRİŞ

Çevremizde meydana gelen fiziki, biyolojik ve sosyal olayları incelemek için izlenen yollardan biri, olayın matematik modelini yapmaktır. Genel anlamda **matematik model**, bir işlemin veya fiziksel sistemin esaslarını belirten ve matematik terimlerle ifade eden bağıntılardır.

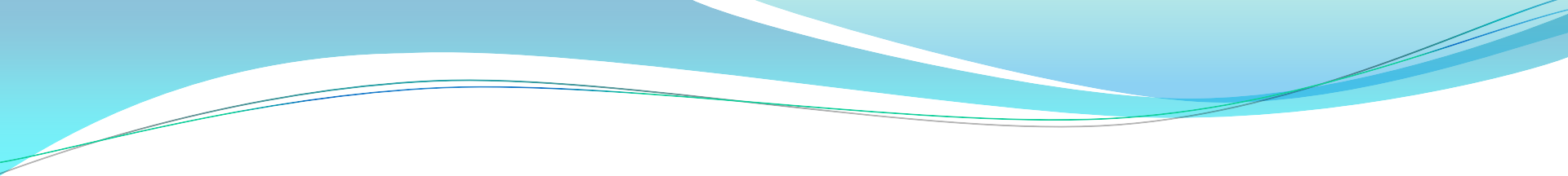


$mgh \rightarrow$ Potansiyel enerji

$\frac{1}{2}mv^2 \rightarrow$ Kinetik enerji

$$mgh = \frac{1}{2}mv^2 \Rightarrow v = \sqrt{2gh}$$

Matematik modellemeye bütün etkiler göz önüne alındığında, modelin çözümlenmesi zor olabilir. Bu sebeple, çözümü basitleştirmek için bazı kabuller yapılır. Mesela, düşen bir cismin hareketi incelenirken hava direncinin ihmal edilmesi gibi.



Bu kabullerden sonra matematiğin verdiği araçlar kullanılarak matematik model çözülür. Çözümde amaç, analitik/kapalı veya kesin çözümün bulunmasıdır. Bu çözüm, sayıların yuvarlatılmasından kaynaklanan hatalar haricinde, matematik modeli kesin olarak sağlar.

Matematik modelin analitik çözümü her zaman mümkün değildir. Böyle durumlarda sayısal çözüm metotlarından faydalanılır. **Sayısal yöntemler**, matematik problemlerinin, aritmetik işlemlerle çözülebilmelerini sağlayan tekniklerdir.

A hand holding a pen is pointing at a background of binary code (0s and 1s) on a blue surface. The background also features some abstract geometric patterns and lines. The text is centered in a light blue rounded rectangle.

Sayısal analizdeki bazı kavramlar

Ardışık yaklaşım (iterasyon): Ardışık yaklaşım veya iterasyon, bir işlemin veya bir hareket şeklinin tekrarı demektir. Sayısal analizde ise bir işlemin tekrar tekrar uygulanmasıdır.

iterasyon sayısı	x	$g(x)$	$h= x_n-x_{n-1} $
1	8	0,054946917	7,945053083
2	0,054946917	2,918701514	2,863754597
3	2,918701514	0,697161304	2,221540209
4	0,697161304	2,117066992	1,419905688
5	2,117066992	1,040892786	1,076174206
6	1,040892786	1,782765652	0,741872867
7	1,782765652	1,230264839	0,552500813
8	1,230264839	1,621707926	0,391443087
9	1,621707926	1,333435008	0,288272918
10	1,333435008	1,540173057	0,206738049
11	1,540173057	1,388919019	0,151254038
12	1,388919019	1,498032798	0,109113779
13	1,498032798	1,418494205	0,079538593
14	1,418494205	1,476043484	

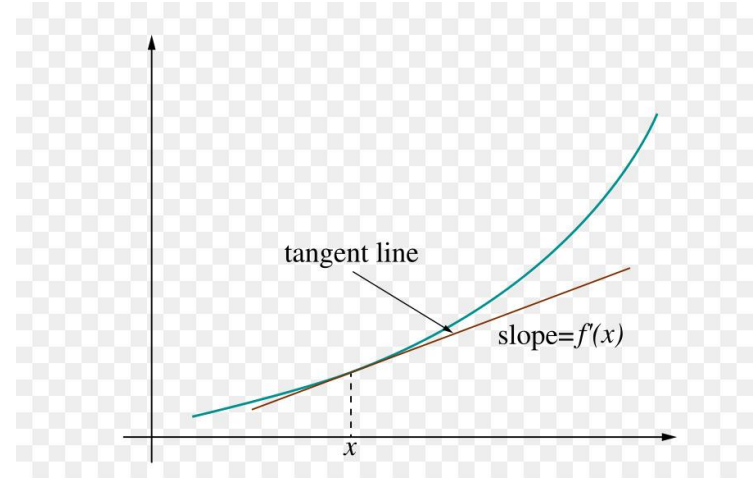
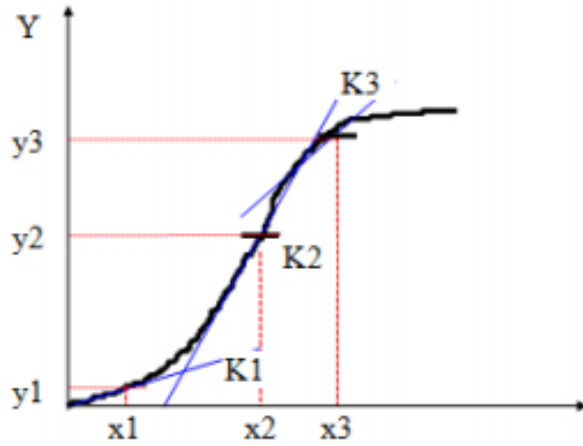
Dijit sayısı: Bir sayının başında bulunan sıfırları saymadan, sayıda bulunan rakam sayısına o sayının dijit sayısı denir. Bir sayının dijit sayısı ile ondalık basamak sayısı farklıdır. Örneğin;

- 0,00156 sayısı 3 dijitli, 5 ondalık basamaklı bir sayıdır.
- 13,78 sayısı ise 4 dijitli, 2 ondalık basamaklı bir sayıdır.

Yuvarlatma ve atma: Bir sayı t . ondalık basamağa yuvarlatılmak istendiğinde, sayının t . basamaktan sonraki parçası $\frac{1}{2} \times 10^{-t}$ den büyük ya da eşit ise t . basamağa 1 eklenir; küçük ise t . basamak olduğu gibi bırakılır. Atma işleminde ise t . basamaktan sonraki kısım olduğu gibi atılır. Tabloda verilen sayıların 3. basamağına yuvarlatma ve atma işlemleri gösterilmiştir.

Sayı	Yuvarlatma	Atma
0,4397	0,440	0,439
0,43750	0,438	0,437
0,43642	0,436	0,436

Doğrusallaştırma (lineerleştirme): Bir fonksiyonun küçük bir bölgede bir doğru olarak kabul edilmesidir. Bu işlem yapılırken, fonksiyon yerine teğeti alınabileceği gibi fonksiyonun iki noktasını birleştiren doğru (kiriş) da alınabilir.



Algoritma: Herhangi bir sorunun çözümü için izlenecek yol anlamına gelmektedir. Çözüm için yapılması gereken işlemler hiçbir alternatif yoruma izin vermeksizin sözel olarak ifade edilir. Diğer bir deyişle algoritma verilerin bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağını, sonucun nasıl ve nereye yazılacağını sözel olarak ifade edilmesi biçiminde tanımlanabilir. Algoritma hazırlanırken, çözüm için yapılması gerekli işlemler, öncelik sıraları gözönünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdırlar.

ÖRNEK: Verilen iki sayının toplamının bulunmasının algoritması aşağıdaki gibi yazılır:

Adım 1-Başla

Adım 2-Birinci sayıyı oku

Adım 3-İkinci sayıyı oku

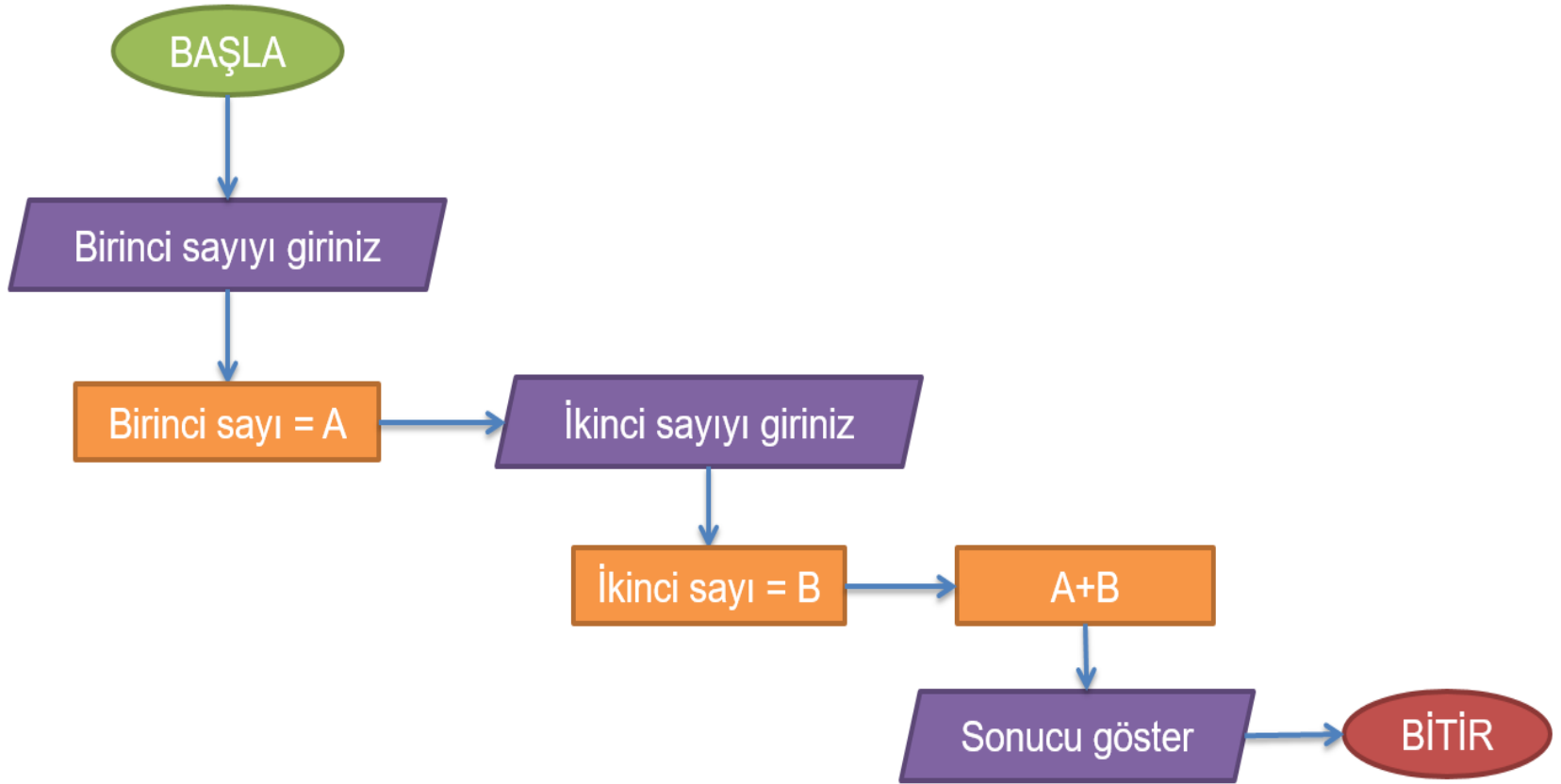
Adım 4-İki sayıyı topla

Adım 5-Dur

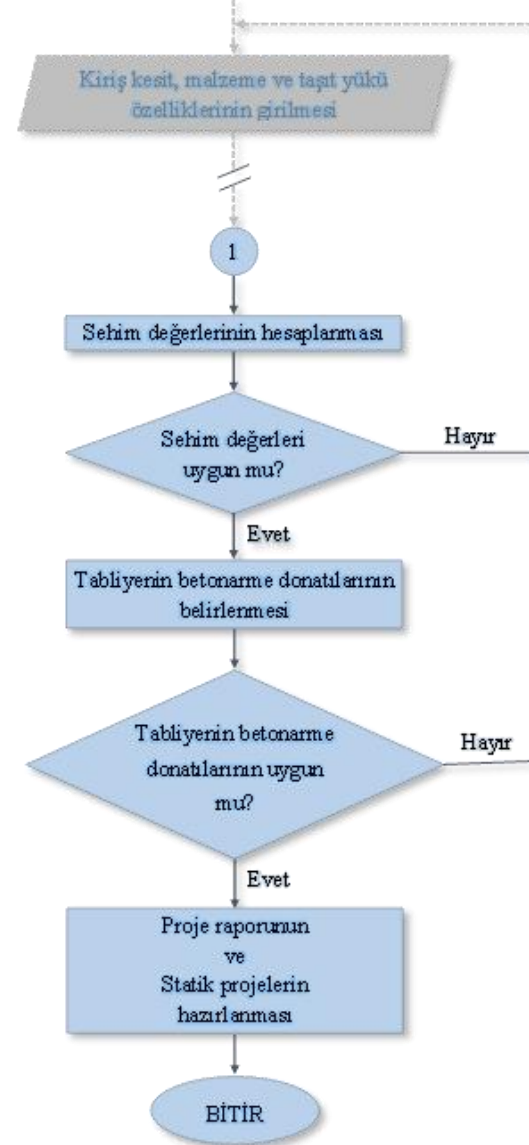
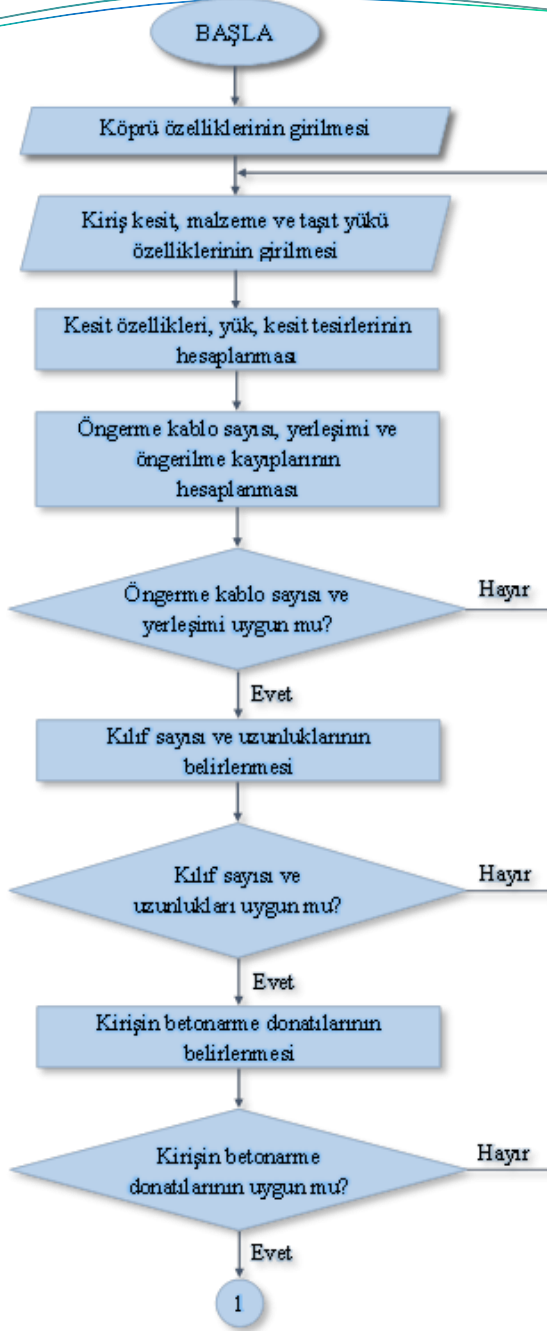
Akış Şemaları (diyagramları): Herhangi bir sorunun çözümü için izlenmesi gerekli olan aritmetik ve mantıksal adımların söz veya yazı ile anlatıldığı algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline "akış şemaları" veya FLOWCHART adı verilir. Akış şemalarının algoritmadan farkı, adımların simgeler şeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösterilmesidir.

Programın saklanacak esas belgeleri olan akış şemalarının hazırlanmasına, sorun çözümlenmesi sürecinin daha kolay anlaşılır biçime getirilmesi, iş akışının kontrol edilmesi ve programın kodlanmasının kolaylaştırılması gibi nedenlerle başvurulur. Uygulamada çoğunlukla, yazılacak programlar için önce programın ana adımlarını(bölümlerini) gösteren genel bir bakış akış şeması hazırlanır. Daha sonra her adım için ayrıntılı akış şemalarının çizimi yapılır.





Akış diyagramı:



Örnek program adımları

```
1  +using ...
21
22  +namespace barbarosICAD ...
73148
73149
73150
```

```
22 namespace barbarosICAD
23 {
24
25
26     public partial class Form1 : Form
27     {
28
29         AnalizDurumuForm frmAnaliz = new AnalizDurumuForm();// analiz durumu penceresi için
30
31
32         List<PointF> cables = new List<PointF>();
33         float ordinate = 0;
34         int cableSize = 0;
35         double baslangicEksantiriste;
36         double Ie;
37
38         public List<Kılıf> kiliflamaUstDegerleri = new List<Kılıf>();
39         public List<Kılıf> kiliflamaAltDegerleri = new List<Kılıf>();
40         public int kilifSayisiCekmeDegeri = 0;
41         public int kilifSayisiBasincDegeri = 0;
42
43
44         public Form1()...
45
46         //***** TUŞLARI KULLANARAK TABLAR ARSI GEÇİŞ*****
47         Proje serit menüsü arası geçişler
180
181         Analiz sonuçları serit menüsü arası geçişler
234
235
236         //Analiz Sonuçları butonuna basınca yeni tabın açılması
237
238
239         private void raporToolStripMenuItem_Click(object sender, EventArgs e)...
240         private void projeToolStripMenuItem_Click(object sender, EventArgs e)...
241
242
243
244
245
246
247         public void asama0()...
248
249         public void asama01()...
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Windows.Forms;
9  //pdf için eklenen using cümlecikleri
10 using System.IO;
11 using CrystalDecisions.CrystalReports.Engine;
12 using CrystalDecisions.Shared;
13 using CrystalDecisions.Windows.Forms;
14 using System.Windows.Forms.DataVisualization.Charting;
15 //AutoCad için
16 using AutoCAD;
17 using System.Threading;
18 using EnvDTE;
19 using EnvDTE80;
20 using System.Drawing.Drawing2D;
21
22 +namespace barbarosICAD ...
73148
73149
73150
```