

Programmeren met de TI-84 Plus

Philip Bogaert

1. Basiscommando's

1.1. De sequentie

probleemstelling 1

Schrijf een programma dat de oppervlakte en de omtrek van een rechthoek berekent nadat je de lengte en breedte hebt ingevoerd.

Analyse

Omtrek & Oppervlakte van een rechthoek
Lees de lengte van de rechthoek Lees de breedte van de rechthoek
$Omtrek = 2 * (lengte + breedte)$
$Oppervlakte = lengte * breedte$
Laat het resultaat verschijnen

Het programma invoeren

Druk **PRGM** ► ► om het menu PRGM NEW op te roepen

Druk **ENTER** om de optie 1:Create New te kiezen. De Name= aanwijzer verschijnt op het scherm en het toetsenbord wordt in alpha-lock gezet. Druk **[R] [E] [C] [H] [T] [H] [K] ENTER** om

```
EXEC EDIT NEW
1:Create New

PROGRAM
Name=RECHTHK

PROGRAM:RECHTHK
:
```

RECHTHK als naam van het programma op te geven.

☞ De naam van een programma is beperkt tot 8 alfanumerieke tekens.

Je bevindt je nu in het programmascherm. De dubbele punt geeft aan dat dit het begin van een programmaregel is.

Het programma moet naar de lengte en de breedte van de rechthoek vragen, dit programmeer je als volgt:

Druk **PRGM** ► I/O en kies de optie 1:Input in het menu.

In de programmaregel wordt Input ingevoerd.

Druk **2nd** [ALPHA] om Alpha-Lock vast te zetten en tik ["] [L] [E] [N] [G] [T] [E] [] ["] **ALPHA** , **ALPHA** [L] **ENTER**

Op dezelfde manier vraag je naar de breedte.

Druk **PRGM** ► I/O en kies de optie 1:Input in het menu.

Druk **2nd** [ALPHA] om Alpha-Lock vast te zetten en tik ["] [B] [R] [E] [E] [D] [T] [E] [] ["] **ALPHA** , **ALPHA** [B] **ENTER**

De lengte wordt nu opgeslagen in de variabele L, de breedte in de variabele B.

☞ Prompt is hetzelfde als Input, alleen kun je nu geen tekst meegeven bij de vraag naar invoer.

Bereken nu de omtrek en de oppervlakte en sla de resultaten op in de variabelen O (= omtrek) en S (= oppervlakte).

Druk **2** * **(** **ALPHA** [L] **+** **ALPHA** [B] **)** **STO>** **ALPHA** [O] **ENTER**
Druk **ALPHA** [L] * **ALPHA** [B] **STO>** **ALPHA** [S] **ENTER**

```
CTL PRGM EXEC
1:Input
2:Prompt
3:Disp
4:DispGraph
5:DispTable
6:Output(
7↓getKey
```

```
PROGRAM:RECHTHK
:Input "LENGTE "
:L
:
```

```
PROGRAM:RECHTHK
:Input "LENGTE "
:L
:Input "BREEDTE "
:B
:
```

```
PROGRAM:RECHTHK
:Input "LENGTE "
:L
:Input "BREEDTE "
:B
:2*(L+B)→O
L*B→S
:
```

Nu moet je nog de uitvoer verzorgen:

Druk **PRGM** ► I/O en kies de optie 3:Disp in het menu.
In de programmaregel wordt Disp ingevoerd.

Druk **2nd** [ALPHA] om Alpha-Lock vast te zetten en tik ["] [O] [M] [T] [R] [E] [K] [] ["] **ALPHA** [] **ALPHA** [O] **ENTER**

Analoog voor de oppervlakte.

Druk **PRGM** ► I/O optie 3:Disp.
Druk **2nd** [ALPHA] en tik ["] [O] [P] [P] [E] [R] [V] [L] [A] [K] [T] [E] [] ["] **ALPHA** [] **ALPHA** [S] **ENTER**

```
PROGRAM:RECHTHK
:2*(L+B)→O
:L*B→S
:Disp "OMTREK ",
O
:Disp "OPPERVLAK
TE",S
:█
```

Je sluit het programma af met **PRGM** optie 7:End, **2nd** [Quit].

De TI-84 procedure

```
PROGRAM:RECHTHK
:Input "LENGTE ",L
:Input "BREEDTE ",B
:2*(L+B) → O
:L*B → S
:Disp "OMTREK ",O
:Disp "OPPERVLAKTE",S
STOP
```

Het programma oproepen

PRGM RECHTHK **ENTER**

ENTER

Lengte 13 **ENTER**
Breedte 7 **ENTER**

```
LENGTE 13
BREEDTE 7
OMTREK
OPPERVLAKTE        40
                         91
                         Done
█
```

Wijzigingen aanbrengen in het programma

Verander het programma, zodat tevens de lengte van de diagonaal wordt berekend.

PRGM ► EDIT optie RECHTHK
ENTER

```
PROGRAM:RECHTHK
:L
:Input "BREEDTE
",B
:2*(L+B)→O
:L*B→S
:█
:Disp "OMTREK ",
```

Je moet nu een berekeningsregel invoegen:

Ga op de regel staan waar de nieuwe regel moet worden ingevoegd en druk **2nd** **[INS]** **[ENTER]**

Bereken de lengte van de diagonaal en sla deze op in de variabele D:

2nd **[√]** **[ALPHA]** **[L]** **[x²]** **+** **[ALPHA]** **[B]** **[x²]** **)** **[STO>]** **[ALPHA]** **[D]**
[ENTER]

En laat dit resultaat op het scherm verschijnen:

Ga naar het einde van het programma,

[PRGM] **[▶]** I/O optie 3:DISP
Druk **2nd** **[ALPHA]** en tik **["]** **[D]** **[|]** **[A]**
[G] **[O]** **[N]** **[A]** **[A]** **[L]** **[]** **["]** **[ALPHA]** **[,]**
[ALPHA] **[D]** **[ENTER]**

2nd **[QUIT]**

```
PROGRAM:RECHTHK
:√(L2+B2)→D
:Disp "OMTREK ",
0
:Disp "OPPERVLAK
TE",S
:Disp "DIAGONAAL
",D
```

Roep het programma op en controleer of je aanvullingen werken !!

```
PRGMRECHTHK
LENGTE 4
BREEDTE 3

OMTREK          14
OPPERVLAKTE    12
DIAGONAAL      5
                Done
```

probleemstelling 2

Schrijf een programma dat de drie hoeken van willekeurig driehoek berekent als de drie zijden gegeven zijn. (ZZZ)

Analyse

Oplossen van een driehoek ZZZ
Lees de drie zijden van de driehoek
$\cos \alpha = \frac{b^2 + c^2 - a^2}{2bc} \rightarrow d$
$\cos \beta = \frac{a^2 + c^2 - b^2}{2ac} \rightarrow e$
$\cos \gamma = \frac{a^2 + b^2 - c^2}{2ab} \rightarrow f$
$\alpha = \cos^{-1}(d)$
$\beta = \cos^{-1}(e)$
$\gamma = \cos^{-1}(f)$
Laat het resultaat verschijnen

Het programma invoeren

Druk **PRGM** ► ► om het menu PRGM NEW op te roepen.

Druk **ENTER** om de optie 1:Create New te kiezen. De Name= aanwijzer verschijnt op het scherm en het toetsenbord wordt in alphaslock gezet. Druk **[D][R][I][E][H][Z][Z][Z]** **ENTER** om DriehZZZ als naam van het programma op te geven.

Je bevindt je nu in het programmascherm.

Het programma moet naar de drie zijden van de driehoek vragen, dit programmeer je als volgt :

PRGM ► I/O 2:Prompt **ALPHA** [A] ,
ALPHA [B] , **ALPHA** [C] **ENTER**

```
PROGRAM:DRIEHZZZ
:Prompt A,B,C
:█
```

Bereken de cosinus-waarden van de hoeken en sla die op in de variabelen D, E en F :

$(\text{ALPHA}[B]^2 + \text{ALPHA}[C]^2 - \text{ALPHA}[A]^2) / (2 \text{ALPHA}[B] \text{ALPHA}[C])$ **STO>** **ALPHA** [D] **ENTER**

...

```
PROGRAM:DRIEHZZZ
:Prompt A,B,C
:(B^2+C^2-A^2)/(2BC)
)→D
:(A^2+C^2-B^2)/(2AC)
)→E
:(A^2+B^2-C^2)/(2AB)
)→F
```

Bereken de hoekgrootten en sla deze op in de variabelen I, J en K :

2nd [cos⁻¹] **ALPHA** [D]) **STO>** **ALPHA** [I] **ENTER**

...

Toon het resultaat in graden, minuten en seconden :

PRGM ► I/O 3:Disp **2nd** [ALPHA] [I] **[H][O][E][K][A]["]** , **ALPHA** [I] **2nd** [ANGLE] 4: ► DMS **ENTER**

...

2nd [QUIT]

```
PROGRAM:DRIEHZZZ
:cos-1(D)→I
:cos-1(E)→J
:cos-1(F)→K
:Disp "HOEK A", I
►DMS
:Disp "HOEK B", J
►DMS█
```

Het programma is af, probeer het even uit :

PRGM DRIEHZZZ **ENTER**

ENTER

A=? 6

ENTER

B=? 7

ENTER

C=? 9

ENTER

```
HOEK A
  41°45'7.939"
HOEK B
  50°58'37.911"
HOEK C
  87°16'14.151"
      Done
```

☞ Vooraleer je het programma start, controleer je in de toekomst best even of je wel in de juiste hoekmodus staat. (**MODE**)

De TI-84 procedure

```
PROGRAM:DRIEHZZZ
:Prompt A,B,C
:(B2+C2-A2)/(2BC) → D
:(A2+C2-B2)/(2AC) → E
:(A2+B2-C2)/(2AB) → F
:cos-1(D) → I
:cos-1(E) → J
:cos-1(F) → K
:Disp "HOEK A",I ► DMS
:Disp "HOEK B",J ► DMS
:Disp "HOEK C",K ► DMS
STOP
```

probleemstelling 3

Los een willekeurige driehoek op als twee zijden en de ingesloten hoek gegeven zijn. (ZHZ)

De TI-84 procedure

```
PROGRAM:DRIEHZHZ
:Prompt A,B
:Input "HOEK C ",K
: $\sqrt{A^2+B^2-2AB\cos(K)}$ →C
: $(B^2+C^2-A^2)/(2BC)$ →D
: $(A^2+C^2-B^2)/(2AC)$ →E
:cos-1(D)→I
:cos-1(E)→J
:Disp "C",C
:Disp "HOEK A",I ▶ DMS
:Disp "HOEK B",J ▶ DMS
STOP
```

```
PrgrmDTIEHZHZ
A=?5
B=?8
HOEK C 57■
```

```
C
6.740094747
HOEK A
38°28'24.027"
HOEK B
84°31'35.973"
Done
```

probleemstelling 4

Los een willekeurige driehoek op als een zijde en de twee aanliggende hoeken gegeven zijn. (HZH)

De TI-84 procedure

```
PROGRAM:DRIEHHZH
:Prompt A
:Input "HOEK B ",J
:Input "HOEK C ",K
:180°-J-K→I
:A sin(J)/sin(I)→B
:A sin(K)/sin(I)→C
:Disp "B",B
:Disp "C",C
:Disp "HOEK A ",I ▶ DMS
STOP
```

```
PrgrmDRIEHHZH
A=?12
HOEK B 52
HOEK C 37
```

```
B
9.457569477
C
7.222880357
HOEK A
91°0'0"
Done
```


probleemstelling 5

Los een willekeurige driehoek op als een zijde, een aanliggende hoek en de overstaande hoek gegeven zijn. (ZHH)

De TI-84 procedure

```
PROGRAM:DRIEHZHH
:Prompt A
:Input "HOEK B ",J
:Input "HOEK A ",I
:180°-I-J→K
:A sin(J)/sin(I)→B
:A sin(K)/sin(I)→C
:Disp "B",B
:Disp "C",C
:Disp "HOEK C",K ▶ DMS
STOP
```

```
PrgrmDRIEHZHH
A=?12
HOEK B 80
HOEK A 55
B
C 14.42673936
10.35861588
HOEK C 45°0'0"
Done
```

1.2. De selectie

probleemstelling 6

Schrijf een programma dat een getal evalueert. Is het getal groter of gelijk aan nul verschijnt de tekst "positief of nul" in het andere geval verschijnt de tekst "negatief".

Analyse

Getallen evalueren	
Lees een getal	
getal >= 0	
JA	NEEN
"positief of nul" verschijnt	"negatief" verschijnt

Het programma invoeren

PRGM ►► NEW ENTER

PROGRAM Name= [G] [E] [T] [A] [L] ENTER

PRGM ► I/O 1:Input

2nd [ALPHA] ["] [G] [E] [E] [F] [] [E] [E] [N] [] [G] [E] [T] [A] [L] [] ["]
ALPHA [] ALPHA [G] ENTER

PRGM 1:If ALPHA [G] 2nd [Test] 4:>= 0 ENTER

PRGM 2:Then ENTER

PRGM ► I/O 3:Disp

2nd [ALPHA] ["] [P] [O] [S] [I] [T] [I] [E] [F] [] [O] [F] [] [N] [U] [L] ["]
ENTER

PRGM 3:Else ENTER

PRGM ► I/O 3:Disp

2nd [ALPHA] ["] [N] [E] [G] [A] [I] [T] [I] [E] [F] ["] ENTER

PRGM 7:End

De TI-84 procedure

```
PROGRAM:GETAL
:Input "GEEF EEN GETAL ",G
:If G>=0
:Then
:Disp "POSITIEF OF NUL"
:Else
:Disp "NEGATIEF"
:End
STOP
```

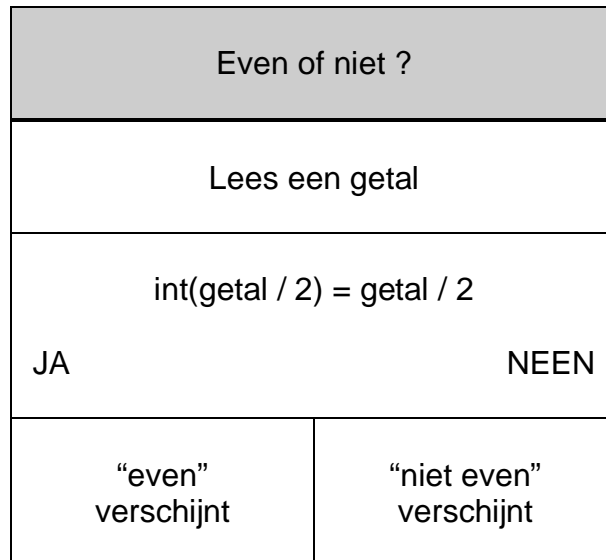


```
prgmGETAL
GEEF EEN GETAL
-17
NEGATIEF
Done
```

probleemstelling 7

Schrijf een programma dat nagaat of een ingelezen getal even is of niet.

Analyse



Het programma invoeren

PRGM ►► NEW ENTER

PROGRAM Name= [E] [V] [E] [N] ENTER

PRGM ► I/O 1:Input

2nd [ALPHA] ["] [G] [E] [E] [F] [] [E] [E] [N] [] [G] [E] [T] [A] [L] [] ["]
ALPHA [] ALPHA [G] ENTER

PRGM 1:If MATH NUM 5:int(ALPHA [G] / 2) 2nd [Test] 1:= ALPHA
[G] / 2 ENTER

PRGM 2:Then ENTER

PRGM ► I/O 3:Disp

2nd [ALPHA] ["] [E] [V] [E] [N] ["] ENTER

PRGM 3:Else ENTER

PRGM ► I/O 3:Disp

2nd [ALPHA] ["] [N] [I] [E] [T] [] [E] [V] [E] [N] ["] ENTER

PRGM 7:End

De TI-84 procedure

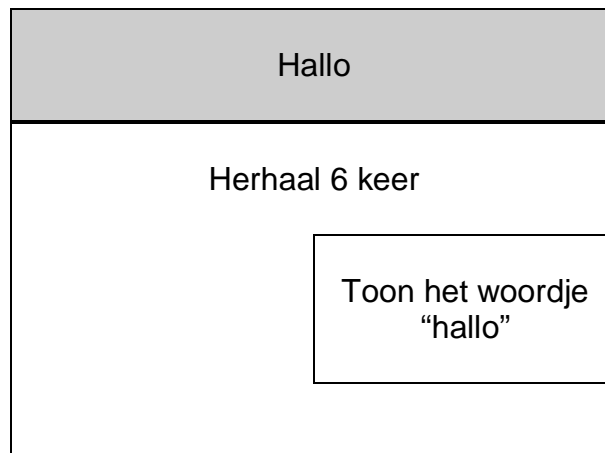
```
PROGRAM:GETAL  
:Input "GEEF EEN GETAL ",G  
:If int(G/2) = G/2  
:Then  
:Disp "EVEN"  
:Else  
:Disp "NIET EVEN"  
:End  
STOP
```

1.3. De begrensde herhaling

probleemstelling 8

Schrijf een programma dat 6 keer het woordje hallo op het scherm laat verschijnen.

Analyse



Het programma invoeren

PRGM ► ► NEW **ENTER**

PROGRAM Name= [H] [A] [L] [L] [O] **ENTER**

Laat de teller i variëren van 1 tot 6 met stapjes van 1.

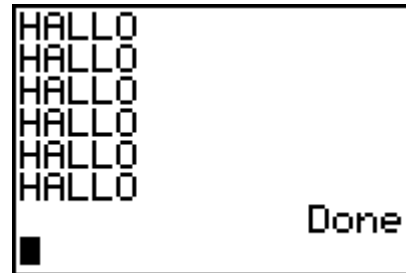
PRGM 4:For(ALPHA [I] , 1 , 6 , 1) ENTER

PRGM ► I/O 3:Disp 2nd [ALPHA] ["] [H] [A] [L] [L] [O] ["] ENTER

PRGM 7:End

De TI-84 procedure

```
PROGRAM:HALLO
:For(I,1,6,1)
  :Disp "HALLO"
:End
```



probleemstelling 9

Schrijf een programma dat de priemgetallen kleiner dan 100 afdruckt.

De TI-84 procedure

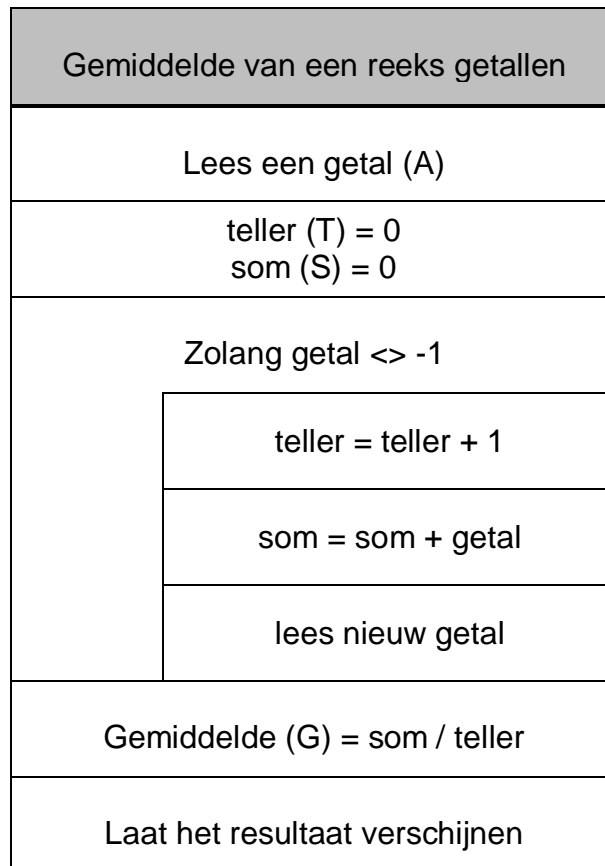
```
PROGRAM:PRIEM
:For (I,2,100)
:1 → P
:For (J,2,√I )
:If fPart(I / J) = 0
:Then
:0 → P
:End
:End
:If P = 1
:Then Disp I
:End
:End
Stop
```

1.4. De voorwaardelijke herhaling

probleemstelling 10

Bereken het gemiddelde van een reeks positieve getallen. De reeks wordt afgesloten door het codegetal -1 .

Analyse



De TI-84 procedure

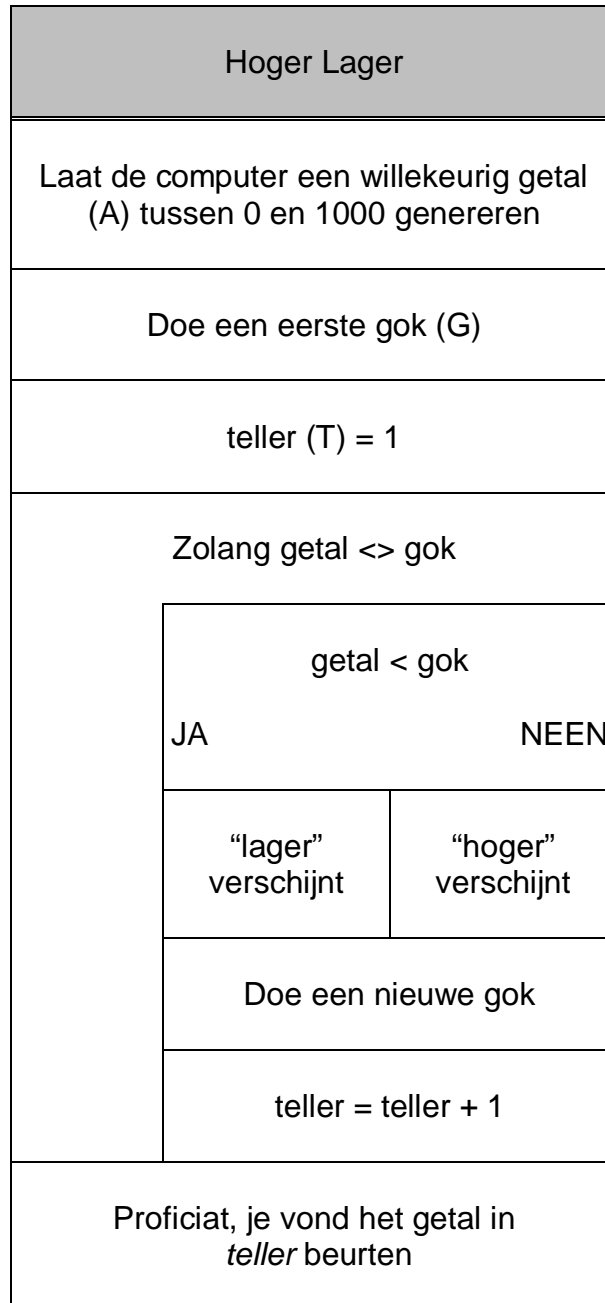
```
PROGRAM:GEMIDDEL
:Input "EERSTE GETAL ",A
:0 → T
:0 → S
:While A ≠ -1
:T+1 → T
:S+A → S
:Input "VOLGEND GETAL ",A
:End
:S/T → G
:Disp "GEMIDDELDE ",G ► Frac
STOP
```

```
PrgrM GEMIDDEL
EERSTE GETAL 5
VOLGEND GETAL 6
VOLGEND GETAL 7
VOLGEND GETAL 8
VOLGEND GETAL 6
VOLGEND GETAL -1
VOLGEND GETAL 8
VOLGEND GETAL 6
VOLGEND GETAL -1
GEMIDDELDE
32/5
Done
```

probleemstelling 11

Ontwerp een spelletje hoger lager.

Analyse



De TI-84 procedure

```
PROGRAM:HOGLAG
:randInt(0,1000) → A
:Input "DOE EEN GOK ",G
:1 → T
:While A ≠ G
:If A<G
:Then
:Disp "LAGER"
:Else
:Disp "HOGER"
:End
:Input "NIEUWE GOK ",G
:T+1 → T
:End
:Disp "PROFICIAT JE VOND HET GETAL IN ",T," BEURTEN"
STOP
```

```
PrgrmHOGLAG
DOE EEN GOK 500
HOGER
NIEUWE GOK 750
HOGER
NIEUWE GOK 825
HOGER
NIEUWE GOK ■
```

```
HOGER
NIEUWE GOK 887
HOGER
NIEUWE GOK 918
LAGER
NIEUWE GOK 902
HOGER
NIEUWE GOK
```

```
NIEUWE GOK 910
HOGER
NIEUWE GOK 914
PROFICIAT JE VO... 8
BEURTEN
Done
■
```


1.5. Menuschermpjes

Wanneer je regelmatig een programma schrijft, heb je al gauw een onoverzichtelijk geheel van procedures waarmee het niet altijd even praktisch werken is.

Het ontwerpen van menuschermpjes biedt hier een oplossing.

Menu(is een instructie waarmee je binnen een programma een sprong kan uitvoeren. Wanneer in de loop van de uitvoering van een programma Menu(wordt ontmoet, wordt het menuscherm getoond met hierin de specifieke menu-opties. Het programma wordt onderbroken totdat je een optie in het desbetreffende menu hebt gekozen.

Syntax :

Menu("titel" , "tekst1", label1, "tekst2", label2, ...)

probleemstelling 12

Maak een menustructuur voor het oplossen van willekeurige driehoeken.

M.a.w. de vier programma's die we ontworpen hebben in het begin, plaatsen we nu als één geheel binnen een menustructuur.

```
PROGRAM: DRIEHOEK
1: ZZZ
2: ZHZ
3: HZH
4: ZHH
```

Het programma invoeren

`PRGM` ► ► NEW `ENTER`

PROGRAM Name= [D] [R] [I] [E] [H] [O] [E] [K] `ENTER`

Ingeven van de menustructuur :

`PRGM` C:Menu(...

`PRGM` 9:Lbl `ALPHA` [A] `ENTER`

```
PROGRAM: DRIEHOEK
: Menu("DRIEHOEK"
: "ZZZ", A, "ZHZ", B
: "HZH", C, "ZHH", D
: )
: Lbl A
:
```

Nu komt de TI-84 procedure die je reeds ontwerpen hebt voor het programma DRIEHZZZ. Dit programma ga je niet herschrijven, maar gewoon kopiëren. Dit doe je als volgt :

Druk `2nd` [RCL]

Druk nu **PRGM** ► ► om het menu PRGM EXEC op te roepen

Kies in dit menu de naam van het programma DRIEHZZZ
Druk **ENTER**. Alle opdrachtregels van het programma DRIEHZZZ
worden nu in het nieuwe programma gekopieerd

Druk **PRGM** F:STOP om het programma te beëindigen.

Herhaal dit om het programma DRIEHZHZ onder het menuitem 2 te krijgen :

PRGM 9:Lbl **ALPHA** [B] **ENTER**

Druk **2nd** [RCL]

Druk **PRGM** ► ► om het menu PRGM EXEC op te roepen

Kies in dit menu de naam van het programma DRIEHZHZ

Druk **ENTER**. Alle opdrachtregels van het programma DRIEHZHZ
worden nu in het nieuwe programma gekopieerd.

Druk **PRGM** F:STOP om het programma te beëindigen.

Analoog voor HZH en ZHH

PRGM 9:Lbl **ALPHA** [C] **ENTER**

Druk **2nd** [RCL]

Druk **PRGM** ► ► om het menu PRGM EXEC op te roepen

Kies in dit menu de naam van het programma DRIEHHZH

Druk **ENTER**. Alle opdrachtregels van het programma DRIEHHZH
worden nu in het nieuwe programma gekopieerd.

Druk **PRGM** F:STOP om het programma te beëindigen.

PRGM 9:Lbl **ALPHA** [D] **ENTER**

Druk **2nd** [RCL]

Druk **PRGM** ► ► om het menu PRGM EXEC op te roepen

Kies in dit menu de naam van het programma DRIEHZHH

Druk **ENTER**. Alle opdrachtregels van het programma DRIEHZHH worden nu in het nieuwe programma gekopieerd.

Druk **PRGM** F:STOP om het programma te beëindigen.

De TI-84 procedure

```
PROGRAM:DRIEHOEK
:Menu("DRIEHOEK","ZZZ",A,"ZHZ",B,"HZH",C,"ZHH",D)
:Lbl A
:Prompt A,B,C
...
:Disp "HOEK A",I ► DMS
:Disp "HOEK B",J ► DMS
:Disp "HOEK C",K ► DMS
:Stop
:Lbl B
...
:Stop
:Lbl C
...
:Stop
:Lbl D
...
:Stop
```

De programma's DRIEHZZZ, DRIEHZHZ, DRIEHHZH en DRIEHZHH zijn nu overbodig geworden, en kunnen dus worden gewist.

1.6. Geheugenbeheer

Geheugenbeheer

Om te controleren of er voldoende geheugen is voor een programma dat je in wil voeren druk je op **2nd** [MEM] om het MEMORY-menu te openen.

Kies

2:Mem Mgmt/Del
7:Prgm

De TI-84 (plus) drukt de geheugenhoeveelheid uit in bytes.

Programma's wissen

Ga naar geheugenbeheer en ga op het te wissen programma staan.

Druk op **DEL** op het programma uit het geheugen te verwijderen. Je krijgt wel nog een melding waarin gevraagd wordt het verwijderen te bevestigen. Kies 2:yes om door te gaan.

```
MEM MGMT
1:About
2:Mem Mgmt/Del...
3:Clear Entries
4:ClrAllLists
5:Archive
6:UnArchive
7↓Reset...
```

```
RAM FREE    20973
ARC FREE    147454
2↑Real...
3:Complex...
4:List...
5:Matrix...
6:Y-Vars...
7↓Prgm...
```

```
RAM FREE    20973
ARC FREE    147454
▷ DRIEHHZH   107
  DRIEHZHH   106
  DRIEHZHZ   135
  DRIEHZZZ   138
  GEMIDDEL   108
  GETAL      78
```

Programma's archiveren

Ga naar geheugenbeheer en ga op het te archiveren programma staan.

Druk op **ENTER** om het programma te archiveren. Links van het programma zal een sterretje verschijnen om aan te geven dat het een gearchiveerd programma betreft.

Om in dit scherm een programma uit het archief te halen, doe je hetzelfde. Ga op het programma staan, druk op **ENTER** en het sterretje verdwijnt.

☞ Een gearchiveerd programma kan niet worden bewerkt of uitgevoerd. Om een gearchiveerd programma te kunnen bewerken of uitvoeren, moet je het eerst uit het archief halen.

Een programma onderbreken

Als je de uitvoering van een programma wil onderbreken, moet je **ON** drukken. Vervolgens verschijnt het menu ERR:BREAK op het scherm.

- Als je wil terugkeren naar het basisscherm, kies je de optie 1:Quit.
- Als je wil overschakelen naar de opdrachtregel waar je programma hebt onderbroken, kies je de optie 2:Goto.

1.7. Communicatieverbindingen tussen twee (gelijke) toestellen

Gegevens selecteren voor verzending (toestel 1)

Druk **2nd** [Link] om het menu LINK SEND op te roepen.

Selecteer het item in het menu dat overeenstemt met het type van gegevens dat je wil verzenden. Het overeenkomstige SELECT venster zal vervolgens op het scherm verschijnen.

Druk **ENTER** om het item te selecteren of deselecteren. De geselecteerde namen worden aangeduid aan de hand van een zwart hokje.

```
SEND RECEIVE
1:All+...
2:All-...
3:Prgm...
4>List...
5:Lists to TI82...
6:GDB...
7↓Pic...
```

```
SELECT TRANSMIT
■ DRIEHHZH PRGM
■ DRIEZHZH PRGM
■ DRIEZHZZ PRGM
■ DRIEZZZZ PRGM
▶ GEMIDDEL PRGM
GETAL PRGM
HALLO PRGM
```

Toestel 2 klaarmaken om te ontvangen

Kies op het ontvangende toestel in het menu LINK RECIEVE de optie 1:Recieve. De melding 'Waiting...' verschijnt op het scherm.

Het toestel staat nu klaar om de doorgezonden gegevens te ontvangen.

Het verzenden van de gegevens (toestel 1)

Kies op het zendende toestel in het menu LINK SEND TRANSMIT de optie 1:Transmit.

Regel na regel verschijnt zowel op het verzendende toestel als op het ontvangende toestel de naam en het gegevenstype van elk overgedragen item.

2. Nulpunten van continue functies

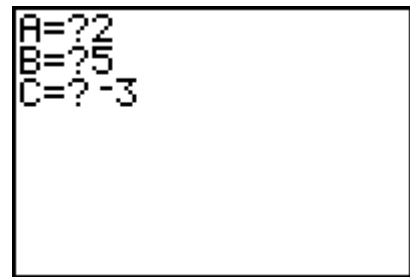
Het bepalen van nulpunten van een functie is in de loop van de geschiedenis van de wiskunde altijd al het onderwerp van ijverige studie geweest.

Een leerling student maakt tijdens zijn wiskundeopleiding kennis met enkele methoden voor het oplossen van vergelijkingen of het bepalen van nulpunten van functies. Deze technieken lenen zich gemakkelijk tot het schrijven van een programma met de GRM. Een overzicht :

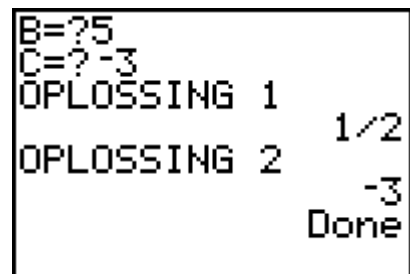
2.1. Bepalen van de reële oplossingen van een VKV

Het programma vraagt naar de drie coëfficiënten van de VKV, en geeft als uitvoer de mogelijke reële oplossingen.

```
PROGRAM:VKV
:Prompt A,B,C
:B2-4AC → D
:If D<0
:Then
:Disp "GEEN OPLOSSINGEN"
:End
:If D=0
:Then
:-B/(2A) → X
:Disp "EEN OPLOSSING",X ► Frac
:End
:If D>0
:Then
:(-B+√(D))/(2A) → X
:(-B-√(D))/(2A) → Y
:Disp "OPLOSSING 1",X ► Frac
:Disp "OPLOSSING 2",Y ► Frac
:End
:Stop
```



```
A=?2
B=?5
C=?-3
```

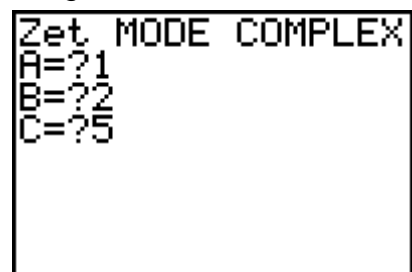


```
B=?5
C=?-3
OPLOSSING 1
OPLOSSING 2
Done
```

2.2. Bepalen van de oplossingen (reëel of complex) van een vierkantsvergelijking

Het programma vraagt naar de drie coëfficiënten van de VKV, en geeft als uitvoer de twee (al dan niet samenvallende) oplossingen.

```
PROGRAM:VKVC
:Disp "Zet MODE COMPLEX"
:Prompt A,B,C
:B2-4AC → D
:(-B-√(D))/(2A) → X
```



```
Zet MODE COMPLEX
A=?1
B=?2
C=?5
```

```

:(-B+√(D))/(2A) → Y
:Disp "OPLOSSING 1",X
:Disp "OPLOSSING 2",Y
:Stop

```

```

B=?2
C=?5
OPLOSSING 1 -1-2i
OPLOSSING 2 -1+2i
Done

```

2.3. Opsporen van gehele nulpunten van een veeltermfunctie

Het programma vraagt naar de veeltermfunctie. Je geeft deze in tussen aanhalingstekens.

Als uitvoer krijg je alle gehele nulpunten van de functie.

```

PROGRAM:NULPUNTZ
:Input "Y1=",Y1
:Y1(0) → A
:If A=0
:Then
:Disp "BRENG X BUITEN HAAKJES"
:Else
:-abs(A) → K
:abs(A) → L
:For(I,K,L,1)
:If Y1(I)=0
:Disp I
:End
:Stop

```

```

Y1="X^3-2X^2-5X+
6"
-2
1
3
Done

```

2.4. Opsporen van rationale nulpunten van een veeltermfunctie

Het programma vraagt naar de veeltermfunctie en de coëfficiënt van de hoogste graadsterm. Als uitvoer krijg je alle rationale nulpunten van de functie.

```

PROGRAM:NULPUNTQ
:Input "Y1=",Y1
:Input "COEFF EERSTE TERM",P
:Y1(0) → A
:If A=0
:Then
:Disp "BRENG X BUITEN HAAKJES"
:Else
:1/P → Q
:-abs(A) → K
:abs(A) → L
:For(I,K,L,Q)
:If abs(Y1(I))<0.00000001
:Disp I ► Frac
:End
:Stop

```

```

Y1="6X^3+7X^2-X-2
COEFF EERSTE TER
6
-1
-2/3
1/2
Done

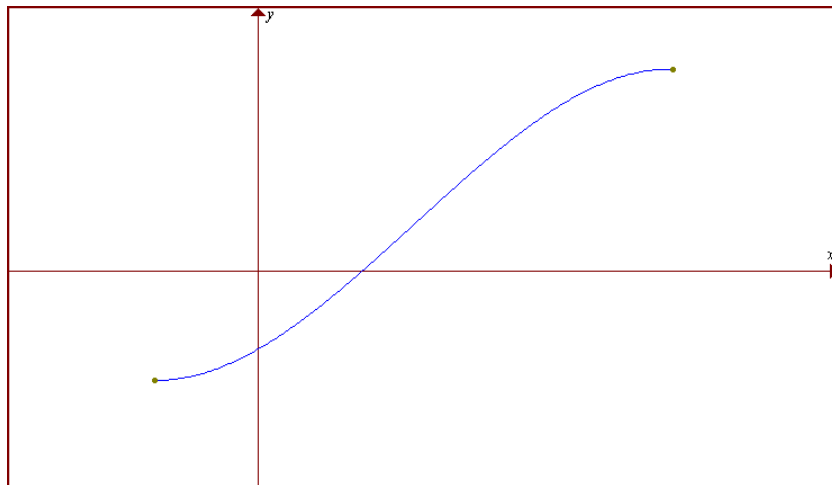
```

3. Numerieke benadering van nulpunten van continue functies

3.1. Methode van Bolzano voor het bepalen van een nulpunt van een continue functie

principe

Als de functie f continu is in $[a,b]$ en $f(a)$ en $f(b)$ hebben een verschillend teken, dan heeft f ten minste één nulpunt $c \in]a,b[$.



programma

De functie geef je vooraf in, in het formule-invoerscherm.

Het programma vraagt om het te zoeken nulpunt links (A) en rechts (B) te schatten. Tevens wordt er ook naar de nauwkeurigheid van het te zoeken nulpunt gevraagd (N).

Als uitvoer krijg je de nieuwe waarden voor A en B, een schatting (M) voor het gevraagde nulpunt en zijn functiewaarde. A en B worden telkens opnieuw berekend totdat de gevraagde nauwkeurigheid bereikt is.

PROGRAMMATEKST	BETEKENIS VAN DE INSTRUCIES
PROGRAM:BOLZANO :Input "A = ",A :Input "B = ",B :Input "NAUWK ",N :While N<(B-A) :(A+B)/2 → M :If Y1(M)=0 :Disp "NULPUNT",M :If Y1(M)*Y1(A)>0 :Then	programmaam geef A, de linkergrens van het startinterval geef B, de rechtergrens van het startinterval geef N, de nauwkeurigheid zolang de nauwkeurigheid onvoldoende is bepaal het midden M van [AB] controleer of de functiewaarde van M nul is zo ja, dan is M het nulpunt controleer of $f(M)$ en $f(A)$ hetzelfde teken hebben

:M → A	zo ja, wordt M de nieuwe A
:Else	zo neen,
:M → B	wordt M de nieuwe B
:End	
:Disp "M = ",M	toon M
:Disp "Y(M) = ",Y1(M)	toon de functiewaarde van M
:Disp "A = ",A	toon A
:Disp "B = ",B	toon B
:Pause	
:End	herhaal de lus (zolang do)
:Stop	

```

Plot1 Plot2 Plot3
\Y1=X^3-2X+5
\Y2=
\Y3=
\Y4=
\Y5=
\Y6=
\Y7=

```

```

prgmBOLZANO
A = -3
B = -2
NAUWK 0.00001

```

```

Y(M) = -2.5
-5.625
A = -2.5
B = -2

```

```

Y(M) = -2.25
A = -1.890625
B = -2.25
-2

```

```

Y(M) = 4.4100664E-6
A = -2.094558716
B = -2.094551086
Done

```

tabel

Stap	Startwaarden		Midden	
	a	b	m	f(m)
1	-3	-2	-2,5	-5,625
2	-2,5	-2	-2,25	-1,890625
3	-2,25	-2	-2,125	-0,345703
4	-2,125	-2	-2,0625	0,351318
5	-2,125	-2,0625		
6				
7				
8				
9				
10				
11				
12				
13				
14				
...				

3.2. Regula Falsi voor het bepalen van een nulpunt van een continue functie

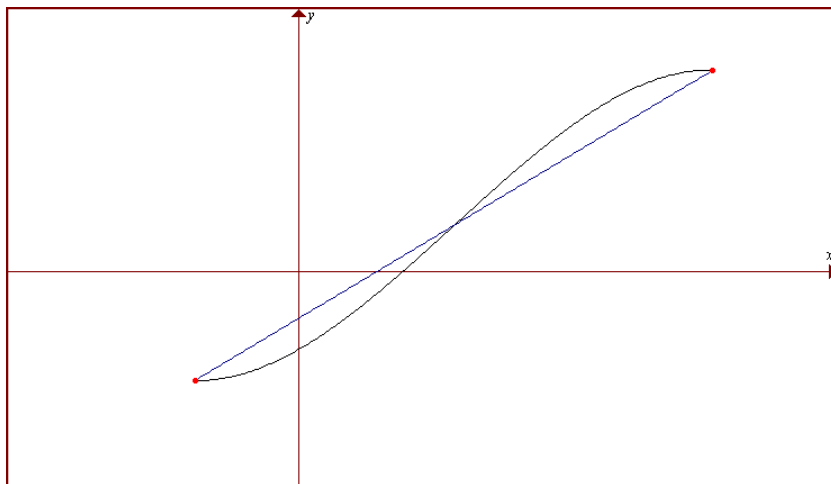
principe

Vertrek opnieuw van twee startwaarden a en b waarvoor $f(a)$ en $f(b)$ een verschillend teken hebben. Vervang nu in $[a,b]$ de grafiek van de functie door de koorde k , d.i. de rechte die de punten $P(a, f(a))$ en $Q(b, f(b))$ verbindt.

$$k = PQ \leftrightarrow y - f(a) = \frac{f(b) - f(a)}{b - a}(x - a)$$

Bepaal het snijpunt $(m, 0)$ van k met de x-as.

$$m = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$



programma

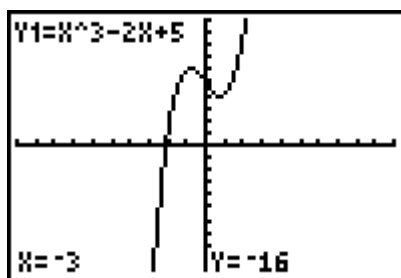
De functie geef je vooraf in, in het formule-invoerscherm.

Het programma vraagt om het te zoeken nulpunt links (A) en rechts (B) te schatten. Tevens wordt er ook naar de nauwkeurigheid van het te zoeken nulpunt gevraagd (N).

Als uitvoer krijg je de nieuwe waarden voor A en B, een schatting (M) voor het gevraagde nulpunt en zijn functiewaarde. A en B worden telkens opnieuw berekend totdat de gevraagde nauwkeurigheid bereikt is.

PROGRAMMATEKST	BETEKENIS VAN DE INSTRUCTIES
PROGRAM:REGULAF :Input "A = ",A :Input "B = ",B :Input "NAUWK ",N	programmamaam geef A, de linkergrens van het startinterval geef B, de rechtergrens van het startinterval geef N, de nauwkeurigheid

<pre> :B-A → K :While N<K :(A*Y1(B)-B*Y1(A))/(Y1(B)- Y1(A)) → M :If Y1(M)=0 :Disp "NULPUNT",M :If Y1(M)*Y1(A)>0 :Then :M-A → K :M → A :Else :B-M → K :M → B :End :ClrHome :Disp "M = ",M :Disp "Y(M) = ",Y1(M) :Pause :End :Stop </pre>	<p>zolang de nauwkeurigheid onvoldoende is bereken M</p> <p>controleer of de functiewaarde van M nul is zo ja, dan is M het nulpunt controleer of f(M) en f(A) hetzelfde teken hebben</p> <p>zo ja, wordt M de nieuwe A zo neen,</p> <p>wordt M de nieuwe B</p> <p>toon M toon de functiewaarde van M</p> <p>herhaal de lus (zolang do)</p>
---	---



```

prgmREGULAF
A = -3
B = -2
NAUWK 0.00001

```

```

M = -2.058823529
Y(M) = .3907999186

```

```

M = -2.08126366
Y(M) = .1472040596

```

```

M = -2.094546951
Y(M) = 5.05686412E-5
Done

```

tabel

Stap	Startwaarden		Midden	
	a	b	m	f(m)
1	-3	-2	-2,058824	0,390800
2	-2,058824	-2	-2,081264	0,147204
3	-2,081264	-2	-2,089639	0,054677
4	-2,089639	-2	-2,092740	0,020203
5	-2,092740	-2		
6				
7				
8				
...				

3.3. Methode van Newton voor het bepalen van een nulpunt van een afleidbare functie

principe

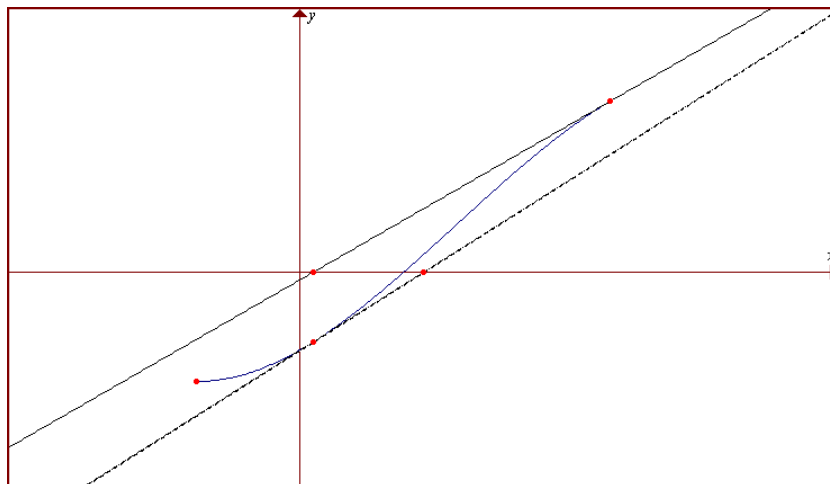
Gegeven een afleidbare functie $f(x)$ en een startwaarde $x = a_1$, die een ruwe benadering is voor het gezochte nulpunt. De vergelijking van de raaklijn t_1 in het punt $P_1(a_1, f(a_1))$ is:

$$y - f(a_1) = f'(a_1) \cdot (x - a_1)$$

Het eerste coördinaatgetal van het snijpunt $(a_2, 0)$ van deze raaklijn met de x-as levert meestal een betere benadering voor het gezochte nulpunt op.

$$a_2 = a_1 - \frac{f(a_1)}{f'(a_1)}$$

Neem nu a_2 als startwaarde en herhaal de hele werkwijze totdat de gevraagde nauwkeurigheid is bereikt.



programma

De functie geef je vooraf in, in het formule-invoerscherm. Het programma vraagt om het te zoeken nulpunt te schatten (A). Tevens wordt er ook naar de nauwkeurigheid van het te zoeken nulpunt gevraagd (N). Als uitvoer krijg je een betere schatting voor het gevraagde nulpunt en zijn functiewaarde. A wordt telkens opnieuw berekend totdat de gevraagde nauwkeurigheid bereikt is.

PROGRAMMATEKST	BETEKENIS VAN DE INSTRUCTIES
PROGRAM:NEWTON :Input "A = ",A :Input "NAUWK ",N	programmamaam geef een startwaarde A geef N, de nauwkeurigheid

<pre> :K=1 :Repeat N>K :A → B :A-Y1(A)/nDeriv(Y1,X,A) → A :abs(B-A) → K :ClrHome :Disp "A = ",A :Disp "Y(A)= ",Y1(A) :Pause :End :Stop </pre>	<p>totdat de nauwkeurigheid voldoende is</p> <p>bereken de nieuwe waarde voor A</p> <p>toon A</p> <p>toon de functiewaarde van A</p> <p>herhaal de lus (totdat)</p>
--	---

<pre> PRGMNEWTON A = -3 NAUWK 0.00001 </pre>	<pre> A = -2.360000026 Y(A) = -3.424256377 </pre>
<pre> A = -2.127196802 Y(A) = -.3711000959 </pre>	<pre> A = -2.094551482 Y(A) = 0 Done </pre>

tabel

Stap	Startwaarden	snijpunt van de raaklijn met de x-as	
	a	a	f(a)
1	-3	-2,360000	-3,424256
2	-2,360000	-2,127197	-0,371100
3	-2,127197	-2,095136	-0,006527
4	-2,095136	-2,094552	-0,000002
5	-2,094552	-2,094551	0

4. Numerieke Integratie

Wanneer je rekenmachine een bepaalde integraal berekent, doet hij dit via een numerieke integratiemethode. Strikt genomen krijg je dan ook maar een benadering van de correcte oplossing.

Hieronder staan een drietal principes van numerieke integratie. Voor elk van hen schrijven we een programma. Je kan de efficiëntie en de nauwkeurigheid van het programma testen aan de hand van de integraal :

$$I = \int_0^8 \frac{4x}{1+x^2} dx = 2 \int_0^8 \frac{d(1+x^2)}{1+x^2} = 2 \left[\ln|1+x^2| \right]_0^8 = 8,34877454$$

De methode die in de TI-84 voorgeprogrammeerd is, is nog een andere, namelijk de *Gauss-Kronrodmethode*.

4.1. Methode van de intervalmiddens

principe

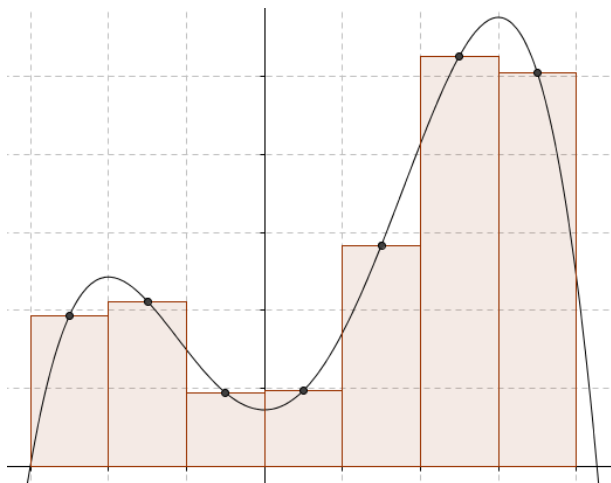
Om $\int_a^b f(x) dx$ te berekenen met de methode van de intervalmiddens verdelen

we $[a, b]$ in n gelijke deelintervallen met breedte $h = \frac{b-a}{n}$.

De oppervlakte tussen de grafiek van f en de x -as binnen een strookje met breedte h wordt benaderd door de oppervlakte van een rechthoek met breedte h en hoogte gelijk aan de functiewaarde van f in het midden van dit interval.

M.a.w. we vervangen de functie f in elk deelinterval door een constante functie.

De som van de oppervlakten van al deze rechthoekjes (Riemannfrietjes) is een benadering voor de bepaalde integraal. Deze benadering zal nauwkeuriger zijn naargelang n groter en h dus kleiner wordt.



formule

Stellen x_1, x_2, \dots, x_n de middens van de deelintervallen voor, en

$f_1 = f(x_1), f_2 = f(x_2), \dots, f_n = f(x_n)$ de functiewaarden in de respectieve middens, dan wordt:

$$\int_a^b f(x) dx = h.f_1 + h.f_2 + \dots + h.f_n = h(f_1 + f_2 + \dots + f_n) \text{ met } h = \frac{b-a}{n}$$

programma:

PROGRAMMATEKST	BETEKENIS VAN DE INSTRUCITIES
PROGRAM:METHINT :ClrHome :Input "A= ",A :Input "B= ",B :Input "N= ",N :(B-A)/N → H :0 → I :For(K,A+H/2,B-H/2,H) :I + Y1(K) → I :End :Disp "BEP.INT. = " :DISP I*H → Frac :Stop	programmanaam wis het scherm geef A, de ondergrens van het interval geef B, de bovengrens van het interval geef N, het aantal deelintervallen H, de breedte van elk deelinterval geef aan de integraal I de beginwaarde 0 begin van de bepaalde herhaling de integraal wordt verhoogd einde van de bepaalde herhaling tekst wordt op het scherm afgedrukt de oplossing verschijnt in breukvorm einde van het programma

```
PROGRAM:METHINT
:ClrHome
:Input "A= ",A
:Input "B= ",B
:Input "N= ",N
:(B-A)/N→H
:0→I
:For(K,A+H/2,B-H
```

```
PROGRAM:METHINT
/2,H)
:I+Y1(K)→I
:End
:Disp "BEP.INT.=
"
:Disp I*H→Frac
:Stop
```

gebruik van het programma:

Vooraleer je het programma uitvoert, moet je het integrandum definiëren in het functiescherm bij Y₁.

Door na de uitvoering van het programma op **ENTER** te drukken, wordt het programma automatisch herstart.

```
Plot1 Plot2 Plot3
\Y1=4X/(1+X^2)
\Y2=
\Y3=
\Y4=
\Y5=
\Y6=
\Y7=
```

```
EDIT NEW
1:METHINT
2:METHSIMP
3:METHTRAP
```

Vul volgende tabel aan:

Aantal deelintervallen (n)	Benadering voor I
4	
10	
20	
50	
100	
200	
500	
1000	
2000	

```
A= 0
B= 8
N= 4
BEP. INT.=
                2944/325
                Done
```

```
A= 0
B= 8
N= 10
BEP. INT.=
                8.474430841
                Done
```

```
A= 0
B= 8
N= 2000
BEP. INT.=
                8.348777246
                Done
```

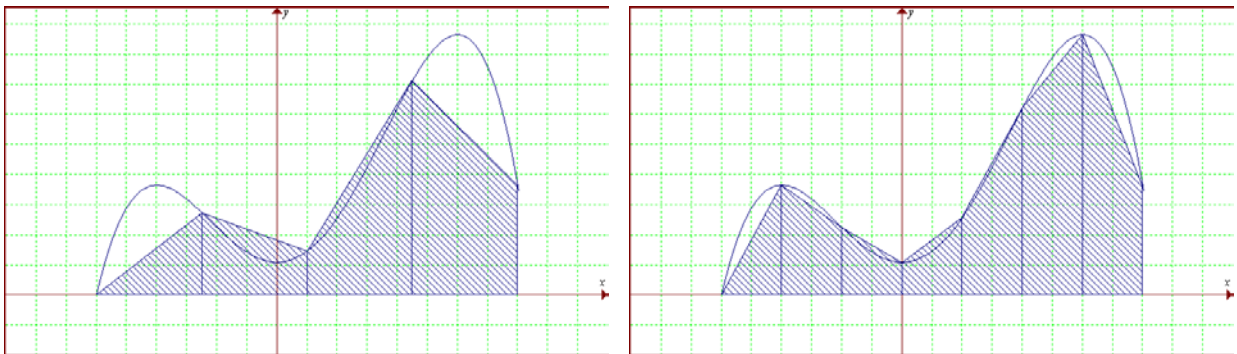

4.2. Trapeziumregel

principe

Om $\int_a^b f(x) dx$ te berekenen met de trapeziumregel verdelen we $[a, b]$ in n gelijke deelintervallen met breedte $h = \frac{b-a}{n}$.

We vervangen de grafiek van f in elk deelinterval door de koorde die de twee uiteinden verbindt, m.a.w. we vervangen de functie f in elk deelinterval door een eerstegraadsfunctie. De oppervlakte tussen de grafiek van f en de x-as binnen een strookje met breedte h wordt benaderd door de oppervlakte van het trapezium gevormd door de twee verticalen $x = x_{i-1}$, $x = x_i$ de x-as en deze koorde.

De som van de oppervlakten van al deze trapeziums is een benadering voor de bepaalde integraal. Deze benadering zal nauwkeuriger zijn naargelang n groter en h dus kleiner wordt.



formule

Stellen $x_0 (= a), x_1, \dots, x_n (= b)$ de grenzen van de deelintervallen voor, en $f_0 = f(x_0), f_1 = f(x_1), \dots, f_n = f(x_n)$ de respectievelijke functiewaarden in deze x-waarden.

De oppervlakte van het eerste trapezium is: $S_1 = \frac{1}{2}h \cdot (f_0 + f_1)$

De oppervlakte van het tweede trapezium is: $S_2 = \frac{1}{2}h \cdot (f_1 + f_2)$

...

De oppervlakte van het laatste trapezium is: $S_n = \frac{1}{2}h \cdot (f_{n-1} + f_n)$

Zodat:

$$\int_a^b f(x) dx \approx S_1 + S_2 + \dots + S_n$$

$$= \frac{1}{2}h(f_0 + 2f_1 + 2f_2 + \dots + 2f_{n-1} + f_n)$$

met $h = \frac{b-a}{n}$

programma:

PROGRAMMATEKST	BETEKENIS VAN DE INSTRUCTIES
PROGRAM:METHTRAP :ClrHome :Input "A= ",A :Input "B= ",B :Input "N= ",N :(B-A)/N → H :0 → I :For(K,A,B,H) :I+Y1(K) → I :End :2*I - Y1(A) - Y1(B) → I :Disp "BEP.INT. = " :DISP I*H/2 → Frac :Stop	programmaam wis het scherm geef A, de ondergrens van het interval geef B, de bovengrens van het interval geef N, het aantal deelintervallen H, de breedte van elk deelinterval geef aan de integraal I de beginwaarde 0 begin van de bepaalde herhaling de integraal wordt verhoogd einde van de bepaalde herhaling alles wordt verdubbeld behalve f(a) en f(b) tekst wordt op het scherm afgedrukt de oplossing verschijnt in breukvorm einde van het programma

```

PROGRAM:METHTRAP
:ClrHome
:Input "A= ",A
:Input "B= ",B
:Input "N= ",N
:(B-A)/N→H
:0→I
:For(K,A,B,H)
:I+Y1(K)→I
:End
:2*I-Y1(A)-Y1(B)→I
:Disp "BEP.INT.= "
:DISP I*H/2→Frac
:Stop
  
```

```

PROGRAM:METHTRAP
:I+Y1(K)→I
:End
:2*I-Y1(A)-Y1(B)
→I
:Disp "BEP.INT.= "
:DISP I*H/2→Frac
  
```

gebruik van het programma:

Vooraleer je het programma uitvoert, moet je het integrandum definiëren in het functiescherm bij Y₁.

Door na de uitvoering van het programma op **ENTER** te drukken, wordt het programma automatisch herstart.

Vul volgende tabel aan:

Aantal deelintervallen (n)	Benadering voor I
4	
10	
20	
50	
100	
200	
500	
1000	
2000	

```
A= 0
B= 8
N= 4
BEP.INT.=
          56192/8177
          Done
```

```
A= 0
B= 8
N= 10
BEP.INT.=
          8.112991206
          Done
```

```
A= 0
B= 8
N= 2000
BEP.INT.=
          8.348769127
          Done
```

4.3. Regel van Simpson of paraboolregel

principe

Om $\int_a^b f(x) dx$ te berekenen met de regel van Simpson verdelen we $[a, b]$ in n gelijke deelintervallen met n even. De breedte van elk deelinterval is $h = \frac{b-a}{n}$.

We vervangen de grafiek van f in twee opeenvolgende deelintervallen door de parabool (met as evenwijdig aan de y -as) die gaat door de drie punten van de grafiek van de functie met als x -waarden de grenzen van de deelintervallen. Dit is mogelijk omdat er een even aantal deelintervallen zijn. M.a.w. we vervangen de functie f in elk deelinterval door een kwadratische functie. De oppervlakte tussen de grafiek van f en de x -as binnen twee opeenvolgende strookjes met breedte h wordt benaderd door de oppervlakte tussen deze parabool en de x -as.

De som van de oppervlakten van alle paraboolstukjes is een benadering voor de bepaalde integraal. Deze benadering zal nauwkeuriger zijn naargelang n groter en h dus kleiner wordt.

formule

Stellen $x_0 (= a), x_1, \dots, x_n (= b)$ de grenzen van de deelintervallen voor, en $f_0 = f(x_0), f_1 = f(x_1), \dots, f_n = f(x_n)$ de respectievelijke functiewaarden in deze x -waarden.

$$\int_a^b f(x) dx \approx \frac{1}{3} h (f_0 + 2(f_2 + f_4 + \dots + f_{n-2}) + 4(f_1 + f_3 + \dots + f_{n-1}) + f_n)$$

$$\text{met } h = \frac{b-a}{n}$$

programma:

PROGRAMMATEKST	BETEKENIS VAN DE INSTRUCTIES
PROGRAM:METHSIMP	programmaam
:ClrHome	wis het scherm
:Input "A= ",A	geef A, de ondergrens van het interval
:Input "B= ",B	geef B, de bovengrens van het interval
:Input "N= ",N	geef N, het aantal deelintervallen
:(B-A)/N → H	H, de breedte van elk deelinterval
:Y1(A)+Y1(B) → I	geef a/d integraal I de beginwaarde $f(a)+f(b)$
:For(K,A+2*H,B-2*H,2*H)	begin van de eerste bepaalde herhaling
:I + 2*Y1(K) → I	de integraal wordt verhoogd
:End	einde van de eerste bepaalde herhaling

<pre> :For(K,A+H,B-H,2*H) :I + 4*Y1(K) → I :End :Disp "BEP.INT. = " :DISP I*H/3 → Frac :Stop </pre>	<p>begin van de tweede bepaalde herhaling de integraal wordt verhoogd einde van de tweede bepaalde herhaling tekst wordt op het scherm afgedrukt de oplossing verschijnt in breukvorm einde van het programma</p>
---	---

<pre> PROGRAM:METHSIMP :ClrHome :Input "A= ",A :Input "B= ",B :Input "N= ",N :(B-A)/N→H :Y1(A)+Y1(B)→I :For(K,A+2*H,B-2 </pre>	<pre> PROGRAM:METHSIMP :For(K,A+2*H,B-2 :H,2*H) :I+2*Y1(K)→I :End :For(K,A+H,B-H,2 :H) :I+4*Y1(K)→I </pre>	<pre> PROGRAM:METHSIMP :I+4*Y1(K)→I :End :Disp "BEP.INT.= " :Disp I*H/3→Frac :Stop </pre>
--	--	---

gebruik van het programma:

Vooraleer je het programma uitvoert, moet je het integrandum definiëren in het functiescherm bij Y_1 .

Door na de uitvoering van het programma op **ENTER** te drukken, wordt het programma automatisch herstart.

Vul volgende tabel aan:

Aantal deelintervallen (n)	Benadering voor I
4	
10	
20	
50	
100	
200	
500	
1000	
2000	

```

A= 0
B= 0
N= 4
BEP.INT.=
7.579503485
Done

```

```

A= 0
B= 0
N= 10
BEP.INT.=
8.368537374
Done

```

```

A= 0
B= 0
N= 2000
BEP.INT.=
8.34877454
Done

```

5. Meetkundige toepassing

Vergelijking van een vlak in de ruimte

Programma voor het herleiden van de determinantvergelijking van een vlak naar de algemene vergelijking $ux + vy + wz + t = 0$.

De determinantvergelijking van een vlak, bepaald door

- één punt en twee stellen richtingsgetallen,
- twee punten en één stel richtingsgetallen,
- drie niet-collineaire punten

zijn respectievelijk:

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ a_1 & b_1 & c_1 & 0 \\ a_2 & b_2 & c_2 & 0 \end{vmatrix} = 0, \quad \begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ a_1 & b_1 & c_1 & 0 \end{vmatrix} = 0, \quad \begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0$$

Noemen we $[A]$ de matrix gevormd door de drie onderste rijen (van de determinanten uit het eerste lid van deze vergelijkingen). We bekomen:

$$\begin{vmatrix} x & y & z & 1 \\ & [A] & & \end{vmatrix} = 0$$

Hierbij is $[A]$ een 3×4 matrix, waarvan de rijen viertallen van de vorm $(x_1 \ y_1 \ z_1 \ 1)$ of $(a_1 \ b_1 \ c_1 \ 0)$ zijn, respectievelijk afgeleid uit een punt $(x_1 \ y_1 \ z_1)$ of een stel richtingsgetallen $(a_1 \ b_1 \ c_1)$ van het vlak α .

Door de matrix $[A]$ te ontrafelen in 4 aparte kolommatrices kan men de cofactoren van de eerste rij van de determinant van de vergelijking berekenen. Door deze determinant te ontwikkelen naar de eerste rij (methode van Laplace) kan men hem gemakkelijk omzetten naar de vorm $ux + vy + wz + t = 0$.

Programma VGLVLAK

- Ontrafel de matrix $[A]$ in 4 aparte kolommatrices

Matr \rightarrow list ($[A]$, L_1 , L_2 , L_3 , L_4)

```
PROGRAM:VLAK
:ClrHome
:Matr→list([A],L
1,L2,L3,L4)
>List→matr(L2,L3
,L4,[B])
>List→matr(L1,L3
,L4,[C])
```

- Definieer 4 (3 x 3) matrices B, C, D en E

List → matrix (L₂ , L₃ , L₄ , [B])
 List → matrix (L₁ , L₃ , L₄ , [C])
 List → matrix (L₁ , L₂ , L₄ , [D])
 List → matrix (L₁ , L₂ , L₃ , [E])

- Bereken det(B), det(C), det(D) en det(E)

det([B]) → B
 det([C]) → C
 det([D]) → D
 det([E]) → E

- Bereken en toon u, v, w en t

Disp "ux + vy + wz + t = 0"
 Pause
 Disp "u", B → Frac
 Disp "v", (-C) → Frac
 Disp "w", D → Frac
 Pause
 Disp "t", (-E) → Frac
 Stop

```
PROGRAM:VLAK
:List▶matr(L1,L2
,L4,[D])
:List▶matr(L1,L2
,L3,[E])
:det([B])→B
:det([C])→C
:det([D])→D
```

```
PROGRAM:VLAK
:det([E])→E
:Disp "UX+VY+WZ+
T=0"
:Pause
:Disp "U",B▶Frac
:Disp "V",(-C)▶F
```

```
PROGRAM:VLAK
:Disp "W",D▶Frac
:Pause
:Disp "T",(-E)▶F
rac
:Stop
:
```

Voorbeeld

$$\alpha \leftrightarrow \begin{vmatrix} x & y & z & 1 \\ -1 & 2 & 0 & 1 \\ 2 & 1 & -2 & 0 \\ -3 & 1 & 2 & 0 \end{vmatrix} = 0$$

Definieer de matrix

$$[A] = \begin{bmatrix} -1 & 2 & 0 & 1 \\ 2 & 1 & -2 & 0 \\ -3 & 1 & 2 & 0 \end{bmatrix}$$

Start het programma VGLVLAK

Direct verschijnen de waarden van u, v, w en t.

$$\alpha \leftrightarrow 4x + 2y + 5z = 0$$

```
[A]
[[-1 2 0 1]
 [2 1 -2 0]
 [-3 1 2 0]]
```

UX+VY+WZ+T=0	:
U	4
V	2
W	5
T	0
	Done