Werktekst 1: Een bos beheren

Berekeningen met rijen op het basisscherm

Op een perceel staan 3000 kerstbomen. Een boomkweker moet beslissen hoeveel bomen er jaarlijks gekapt kunnen worden en hoeveel nieuwe aanplant er nodig is. Hij zal niet alle bomen in één keer kappen want dan heeft hij de eerstvolgende jaren geen opbrengst. Hij besluit elk jaar 10% van de bomen te kappen en er dan weer 450 aan te planten. Hij plant dus meer dan hij kapt om zijn opbrengst op termijn te verhogen. Op het perceel is namelijk plaats voor 5000 bomen.

1. Hoeveel bomen staan er één jaar later op het perceel? En twee jaar later?
2. Onderzoek hoe het aantal bomen op dit perceel de volgende twintig jaar evolueert. Maak hierbij gebruik van de `ANS`-knop van je rekentoestel.
3. Kan de boomkweker zijn beleid blijven voortzetten of staat het perceel na een tijd vol?
4. Op een gegeven moment lijkt er een evenwicht te ontstaan. Hoeveel bomen staan er dan op het perceel?

Tabellen en grafieken

Het aantal bomen B hangt af van de tijd t (in jaren). We volgen de evolutie van het aantal bomen jaar na jaar. We laten de veranderlijke t alleen natuurlijke getallen als waarden aannemen. Zo krijgen we een *rij van getallen* die de evolutie van het aantal bomen beschrijft. Het is niet eenvoudig om een *expliciete vergelijking* te vinden voor deze rij. We kunnen de rij echter wel op een andere manier beschrijven, namelijk aan de hand van een *recursieve vergelijking* (synoniemen: *differentievergelijking*, *recurrente betrekking*) met beginvoorwaarde. Deze recursievergelijking drukt $B(t)$ uit in functie van $B(t-1)$.

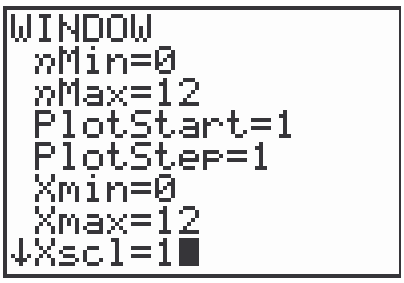
5. Geef deze recursieve vergelijking en beginvoorwaarde.

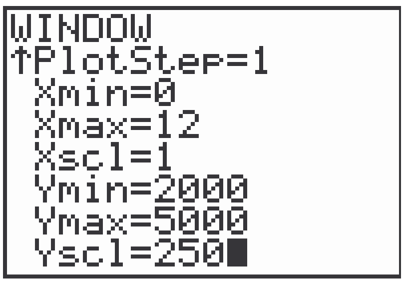
Je kunt recursieve voorschriften ook in je rekentoestel invoeren. Hiervoor moet je eerst de juiste `MODE` instellen. Druk op de `[MODE]`-toets en kies op de vierde regel `SEQ` (van ‘sequence’, het Engels voor rij).

Als je nu de `[Y=]` - toets indrukt, verschijnt in het venster o.a. ‘$u(n)$ =’ in plaats van het bekende ‘Y1=’.

6. Vul op de plaats van ‘`u(`n`)=`’ de recursieve vergelijking in. Voor de rekenmachine wordt de veranderlijke B dus u en de veranderlijke t wordt n. u vind je bij `[2nd][u]` en de veranderlijke n verschijnt bij de `[X,T,θ,n]`-toets. Stel ook de beginwaarde van n in (`nMin`) en de beginwaarde van u (`u(nMin)`).

Nu kun je een tabel en een grafiek van de recursieve vergelijking laten maken. Voor de grafiek moet je nog een goed tekenvenster kiezen. Hieronder zie je hoe dit gebeurt.





Met *n*`Min` en *n*`Max` geef je aan welke termen van de rij *berekend* zullen worden. `PlotStart` en `PlotStep` gebruik je om aan te geven welke termen *getekend* zullen worden. Met `PlotStart` stel je in vanaf welk element van de rij het tekenen moet starten. Omdat we $u(0)$, het *eerste* element van de rij, ook willen laten tekenen, staat `PlotStart` op 1. Omdat we alle termen van de rij willen laten tekenen , staat `PlotStep` op 1. Met `[GRAPH]` wordt de grafiek van de recursieve vergelijking gemaakt. Met `[TRACE]` kun je, zoals bij functies, 'over de grafiek lopen'. De instellingen voor de tabel kun je aanpassen met `[2nd] [TBLSET]`. De tabel wordt gemaakt als je op `[2nd] [TABLE]` drukt.

7. Maak een tabel en een grafiek van de evolutie van het aantal kerstbomen en controleer hiermee de bevindingen die je eerder deed (om de evenwichtswaarde te controleren zul je eerst *n*Max moeten verhogen)

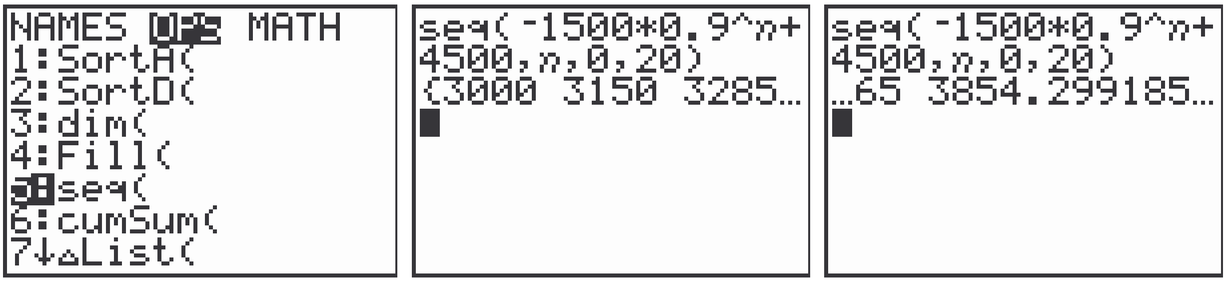
Men kan aantonen (met zeer elementaire middelen, zie b.v. Uitwiskeling 20/3) dat het expliciete voorschrift van de rij gegeven wordt door

$$B(t) = -1500 \cdot 0.9^t + 4500 .$$

8. Voer dit expliciete voorschrift in in je rekenmachine voor de rij $v(n)$ (zonder de beginwaarde te specificeren) en controleer aan de hand van de tabel en de grafiek dat de rijen $u(n)$ en $v(n)$ gelijk zijn.

LIST-commando's

In het LIST-OPS-menu (OPS staat voor 'operations'; `[2nd] [LIST] OPS`) vind je het commando '`seq(`'. De onderstaande schermafdruk toont hoe je hiermee een lijst kunt maken met (een eindig aantal) termen van de rij. M.b.v. het 'pijltje rechts' kun je de volgende termen van de rij zien.



9. Maak een lijst met het aantal bomen dat jaar na jaar gekapt wordt in de eerste 30 jaar.
10. Hoeveel bomen werden er in het totaal gekapt gedurende de eerste 30 jaar? Maak gebruik van het commando '`sum(`'dat je in het LIST-MATH-menu vindt en waarmee je de som kunt berekenen van alle getallen in een lijst.
11. Bekijk hoe de verschillen tussen opeenvolgende aantallen gekapte bomen evolueren. Maak hiervoor gebruik van het commando '`ΔList(`' uit het LIST-OPS-menu, waarmee je de verschillen tussen opeenvolgende elementen van een lijst kunt berekenen.

Uitbreiding: meer in verband met het evenwicht

12. Betekent het bereiken van een evenwicht dat er niets meer verandert?
13. Een evenwicht betekent dat het aantal bomen niet meer verandert. Gebruik dit om het evenwicht op een andere manier te berekenen.
14. Op het moment dat de kleinzoon de zaak overneemt, staan er 4500 bomen op het perceel. Hij houdt dezelfde politiek aan als zijn vader en grootvader: jaarlijks 10% van de bomen kappen en 450 nieuwe bomen planten. Hoe evolueert het aantal bomen op zijn perceel?
15. Door een ongeval kan de kleinzoon in een bepaald jaar slechts 400 nieuwe bomen aanplanten. Daardoor raakt het systeem tijdelijk uit evenwicht. De kleinzoon blijft echter bij zijn werkwijze. Hoe evolueert het aantal bomen?

16. De achterkleinzoon neemt de zaak over. Hij heeft bedenkingen bij de handelswijze van zijn voorvaderen. Niet het hele perceel wordt benut. Er is immers plaats voor 5000 bomen. Kun je er voor zorgen dat het evenwicht op 5000 komt te liggen door
    a. een verandering aan te brengen in het aantal nieuwe bomen dat aangeplant wordt (en verder alles ongewijzigd te laten)
    b. een verandering aan te brengen in het percentage dat gekapt wordt (en verder alles ongewijzigd te laten).

Werktekst 2: Een ander discreet, dynamisch marktmodel

In dit nieuwe discrete, dynamische marktmodel reageren raag en aanbod ogenblikkelijk op de prijs: d.w.z. $v_n = 150 - 5p_n$ en $a_n = -30 + 4p_n$ voor alle *n*. We gaan er niet meer van uit dat er evenwicht is tussen vraag en aanbod, m.a.w. de vergelijking $v_n = a_n$ is niet langer geldig. Als vraag en aanbod op een zeker tijdstip van elkaar verschillen, dan zal de prijs op het volgende tijdstip aangepast worden en met name zo dat het prijsverschil evenredig is met het verschil tussen vraag en aanbod met evenredigheidsfactor 0.02.

1. Veronderstel dat $p_0 = 25$.
   a. Bepaal a_0 en v_0.
   b. In welke richting zal de prijs evolueren?
   c. Met hoeveel zal de prijs wijzigen?
   d. Bepaal p_1.
2. Zelfde vraag, maar nu vertrekkend vanaf p_1.
3. Stel de recursievergelijking op van dit discrete dynamische marktmodel.
4. Gebruik de recursievergelijking (en je grafische rekenmachine) om zeer snel de eerste 10 waarden van de prijs te berekenen.
5. Maak een (TIME-)grafiek van de prijsevolutie.
6. Beschrijf de prijsevolutie in woorden.
7. Maak ook een WEB-grafiek van de prijsevolutie.
8. Wat zou er veranderen als we een andere evenredigheidsfactor zouden nemen?