
Oefening 1

We gaan de Black-Scholes PDE in het discretisatie rooster (s_i) beschouwen met volgende notatie.

$$\begin{aligned}u', u'_j &= u_t, u_t(s_j, t) \\ c^2, c^1, c^0 &= \frac{1}{2}\sigma^2 s^2, rs, r\end{aligned}$$

Benaderingen van u zullen aangeduid worden met U .

De Black-Scholes vergelijking met deze notatie wordt:

$$u' = c_2 u_{ss} + c_1 u_s - c_0 u.$$

Dit zijn tweede orde benaderingen voor de afgeleiden van u (tweede orde centraal):

$$\begin{aligned}u_s &= \frac{u(s+h) - u(s-h)}{2h} + O(h^2) \\ u_{ss} &= \frac{u(s+h) - 2u(s) + u(s-h)}{h^2} + O(h^2)\end{aligned}$$

Hiermee wordt de benaderde Black-Scholes vergelijking in het rooster:

$$U'_j = \frac{c_j^2}{h^2}(U_{j+1} - 2U_j + U_{j-1}) + \frac{c_j^1}{2h}(U_{j+1} - U_{j-1}) - c_j^0 U_j.$$

Dit kunnen we eenvoudiger schrijven in matrix vorm. Voer eerst volgende notatie in

$$D_j = \left[\left(\frac{c_j^2}{h^2} + \frac{c_j^1}{2h} \right), \left(-2\frac{c_j^2}{h^2} - c_j^0 \right), \left(\frac{c_j^2}{h^2} - \frac{c_j^1}{2h} \right) \right].$$

Hiermee wordt dit

$$U'_j = D_j (U_{j-1}, U_j, U_{j+1})^T \text{ voor } 1 \leq j \leq m.$$

Nog meer notatie

$$U' = (U'_1, U'_2, \dots, U'_m)^T$$

$$U_d = (U_0, U_1, \dots, U_{m+1})^T$$

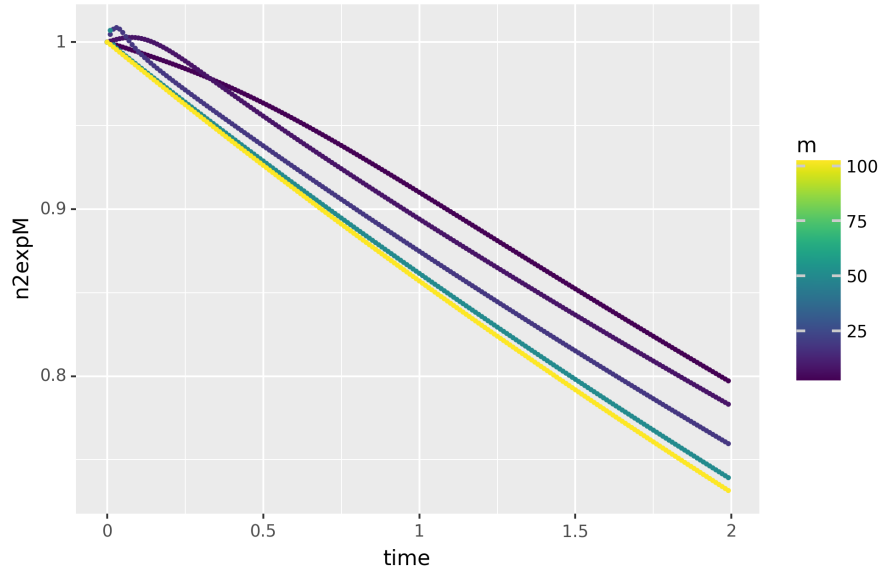
$$A_d = \text{Diag}(D_j)$$

$$U = U_d \text{ zonder de eerste en laatste rij}$$

$$A = A_d \text{ zonder de eerste en laatste kolom}$$

Hiermee wordt dit

$$U' = A_d U_d.$$



Figuur 1: $\|e^{At}\|_2$ vs t voor verschillende m

Door de beginvoorwaarden zijn we enkel geïnteresseerd in U . Door de structuur van A_d en de beginvoorwaarden kunnen we U_0 en U_{m+1} wegwerken in een term $g(t)$.

$$U' = AU + (D_1)_0 U_0 e_1 + (D_m)_2 U_{m+1} e_m.$$

Merk op dat in deze opgave $U_0 = 0$.

Deze notatie wordt geïmplementeerd in `../scr/Model.py`

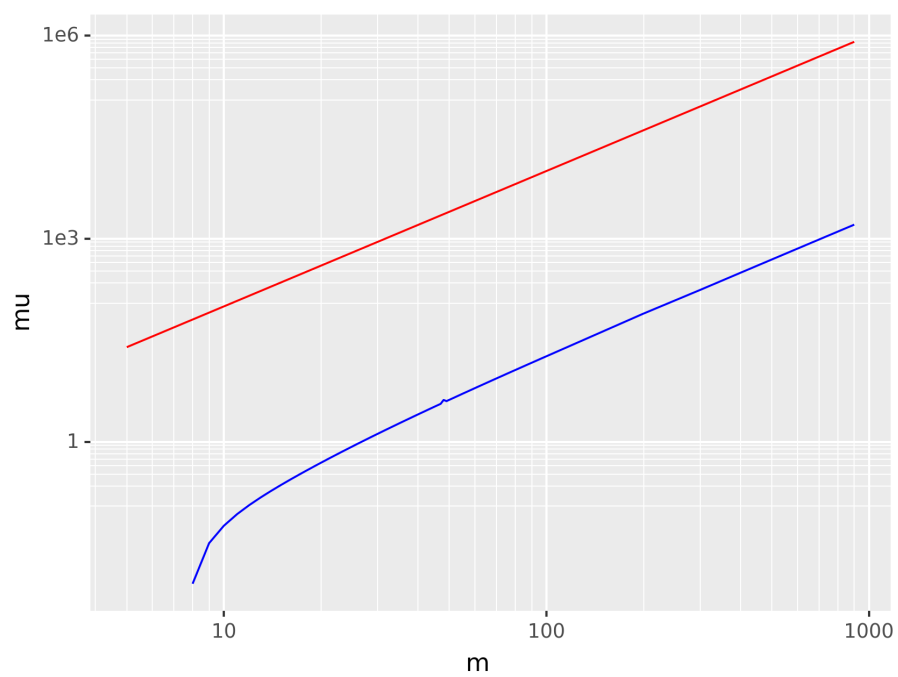
Oefening 2

Kijk naar figuur1 voor de gevraagde grafiek. Dit suggereert dat $\omega = 0$ en $K = 2$ stabiliteit constanten zijn voor deze semi discretisatie.

Als $m \rightarrow \infty$ dan wordt de bijhorende grafiek waarschijnlijk een rechte.

Oefening 3

Kijk naar figuur2 voor $\mu_2(A)$ voor verschillende m . $\mu_2(A)$ groeit even snel als m^2 (kijk grafiek). Dit kan waarschijnlijk bewezen worden met μ_1, μ_2 afschatting. Dit betekent dat voor K de stabiliteit constante bij $T = \infty : K > 1$.



Figuur 2: $(\mu_2(A), m^2)$ vs m

Oefening 4

In dit geval is de impliciete trapezium regel:

$$U(t_{n+1}) = U(t_n) + \frac{1}{2}\tau(AU(t_n) + g(t_n) + AU(t_{n+1}) + g(t_{n+1})).$$

Verzamel $U(t_{n+1})$ (omdat die onbekend is) dit ziet er zo uit:

$$(I - \frac{\tau}{2}A)U(t_{n+1}) = U(t_n) + \frac{\tau}{2}(AU(t_n) + g(t_n) + g(t_{n+1})).$$

Dit is een ijl stelsel die een unieke oplossing heeft voor kleine τ . LU+backsubben is hier de meest voor de hand liggende methode omdat we dit stelsel herhaaldelijk moeten oplossen en niet afhangt van de tijd.

Oefening 5

De implementatie is te vinden in `../scr/DOcall_numer.py` (deze convergeert niet naar de exacte oplossing en werkt niet voor sommige m waarden bv 200.)

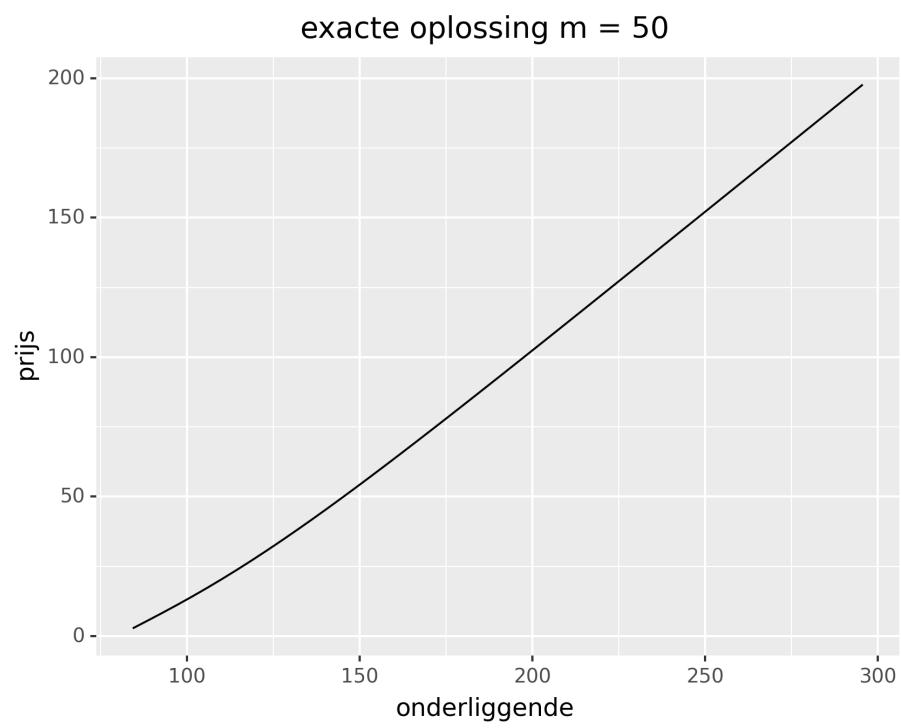
Oefening 6

De implementatie is te vinden in `../scr/DOcall_exact.py`.

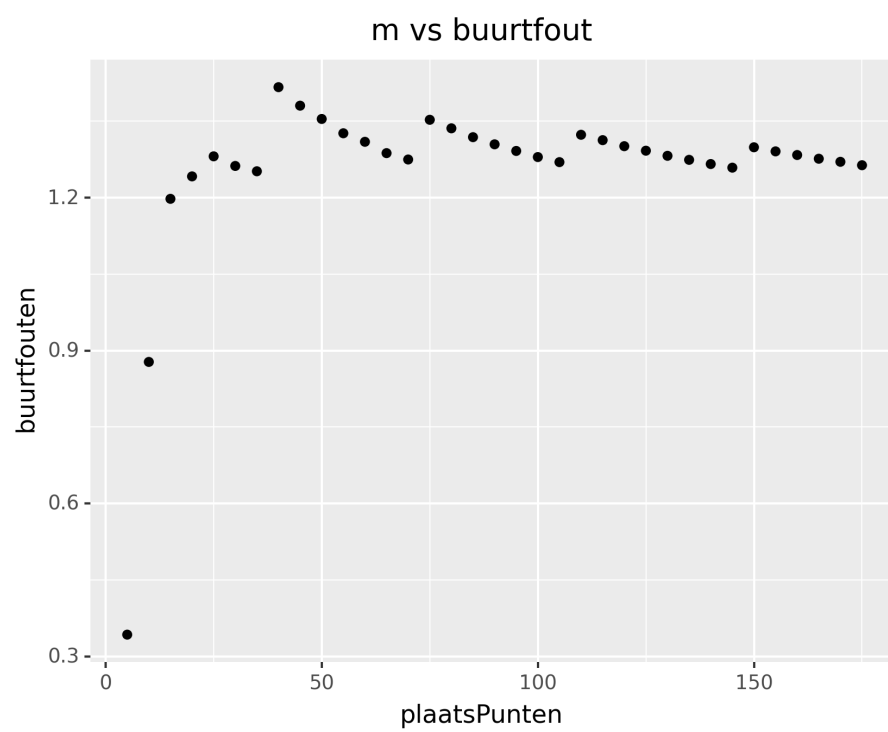
Oefening 7

De numerieke oplossing convergeert niet naar de exacte om een of andere reden. Daarom wordt hier ook een numeriek buurt fout uitgerekend. De implementatie is te vinden in `../scr/foutanalyse.py`.

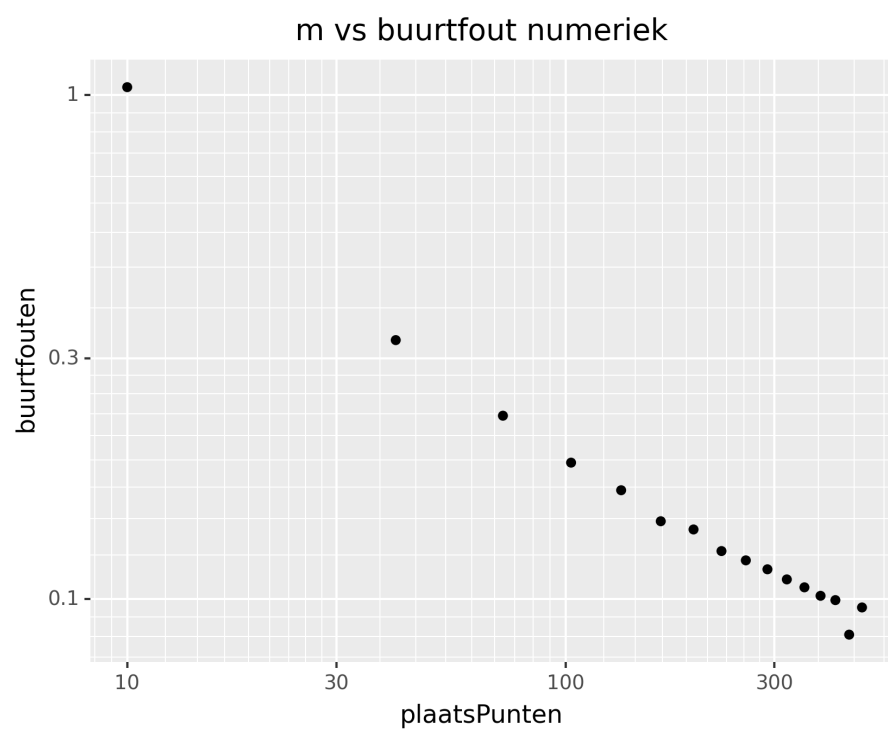
Oefening 8



Figuur 3: exacte oplossing $m = 50$



Figuur 4: ok



Figuur 5: ok