Een fysisch systeem voor de computer

Derk Rouwhorst en Harmen Stoppels

5 januari 2011

Beste lezer, dit is nog niet af, wees gewaarschuwd

Dit verslag is nog lang niet af, maar het is al wel een opzetje. Er moeten nog wel honderden plaatjes ter verduidelijking bij!

Inhoudsopgave

1 Vectoren

We drukken in ons hele profielwerkstuk snelheden, posities en krachten uit in vectoren. Hiermee maken we het rekenwerk een stuk makkelijker en kunnen we formules veel korter opschrijven. We maken gebruik van tweedimensionale vectoren, die uit een x- en een y-component in het cartesisch coördinatenstelsel bestaan. In de voorbeelden en definities in dit onderdeel gebruiken we ook alleen tweedimensionale vectoren, ook al gelden veel ervan ook voor n-dimensionale vectoren.

1.1 Notatie

De namen van vectoren schrijven we dikgedrukt en de componenten boven elkaar met ronde haken eromheen. Een voorbeeld hiervan is $\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 5 \\ -3 \end{pmatrix}$. Dit stelt een vector voor die 5 naar rechts is gericht en 3 naar onder.

1.2 Norm van een vector

De norm van een vector is de lengte of grootte. Deze wordt aangeduid met een dubbele verticale streep rondom de vector. Oftewel, de norm van vector $\mathbf{v}$ is $\|\mathbf{v}\| = \sqrt{({v_1}^2 + {v_2}^2)}$

1.3 Het inwendig product

Het inwendig product van twee vectoren is de som van alle componenten met elkaar vermenigvuldigd. Een voorbeeld hiervan is:

$$\mathbf{v} = \begin{pmatrix} 5 \\ -3 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$
$$\mathbf{v} \cdot \mathbf{u} = 5 \times 4 - 3 \times 2 = -14$$

Het resultaat van die vermenigvuldiging en optelling is het best te zien als hoeveel de kortste vector van de twee in het domein van de langste zit. Deze eigenschap is handig te gebruiken bij het kijken of twee bewegende cirkels elkaar op een bepaald tijdsinterval raken.

Het bewijs voor deze eigenschap maakt gebruik van de cosinusregel. Gegeven de vectoren $a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ en $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ met hetzelfde beginpunt en een onderlinge hoek γ en lijn c tussen beide uiteinden van de vectoren, geldt er:

$$
\begin{aligned}
c^2 &= \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2 \cdot \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\gamma \\
c &= \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \\
(a_1 - b_1)^2 + (a_2 - b_2)^2 &= \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2 \cdot \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\gamma \\
{a_1}^2 + {a_2}^2 + {b_1}^2 + {b_2}^2 - 2a_1b_2 - 2a_2b_2 &= {a_1}^2 + {a_2}^2 + {b_1}^2 + {b_2}^2 - 2 \cdot \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\gamma \\
-2a_1b_2 - 2a_2b_2 &= -2 \cdot \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\gamma \\
-2(\mathbf{a} \cdot \mathbf{b}) &= -2 \cdot \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\gamma \\
\mathbf{a} \cdot \mathbf{b} &= \|\mathbf{a}\| \cdot \|\mathbf{b}\| \cdot \cos\gamma
\end{aligned}
\tag{1}
$$

Met het resultaat hiervan kunnen we later geometrische berekeningen doen, zonder de voor de computer “tijdrovende” functie cosinus of sinus te gebruiken.

1.4 Eenheidsvector

Een eenheidsvector is een vector waarvan de lengte 1 is. Het handige van zo’n vector is dat hij vermenigvuldigd met een getal zijn richting behoudt en de lengte van dat getal aanneemt. Stel je hebt richting waarin een voorwerp beweegt als eenheidsvector, dan geeft vermenigvuldiging met de snelheid de snelheid als vector. De notatie van een eenheidsvector wordt gedaan met een dakje erop: $\|\hat{\mathbf{a}}\| = 1$.

2 Tijdstippen van botsingen

2.1 Cirkel tegen cirkel

Kijken of twee cirkels elkaar raken is theoretisch erg makkelijk te doen. Het komt er simpelweg op neer dat de cirkels elkaar raken als de som van de stralen groter of gelijk is aan de afstand tussen de middelpunten. Het wordt echter lastiger als de cirkels snel bewegen en elkaar tussen twee frames raken. Het probleem is dat het gros van de botsingen tussen cirkels juist tussen twee frames gebeurt.

2.1.1 Formule

Bij het berekenen van de nieuwe posities van de ballen in het volgende frame, moeten we dus als eerste kijken of ze ondertussen botsen, als tweede berekenen wanneer dat is en als derde uitvinden wat voor gevolg die botsing heeft.

Om de uiteindelijke vergelijking te versimpelen gaan wij ervan uit dat elk voorwerp tussen twee frames een constante snelheid heeft en over een rechte lijn beweegt. Dit klopt niet met de realiteit, maar omdat de tijd tussen twee frames erg klein is, maakt het niet zo veel uit.

De cirkels kunnen elkaar op het tijdsinterval tussen twee frames hoogstens twee keer raken (als je geen rekening houdt met het effect van de botsing), namelijk wanneer ze voor het eerst tegen elkaar aankomen en wanneer ze door elkaar heen zijn gevlogen. De botsing die wij moeten vinden is de eerste, de tweede vindt natuurlijk nooit plaats.

Allereerst noemen we de tijd in het eerste frame $t = 0$ en in het tweede frame $t = 1$. We zoeken een waarde van t waarvoor geldt dat $0 \leq t < 1$ en waarbij de afstand tussen de middelpunten gelijk is aan de som van de stralen.

De posities van de ballen geven we aan met vectoren. De nieuwe positie bestaat uit de oude positie met daarbij opgeteld de snelheidsvector vermenigvuldigd met de tijd. Wat we krijgen is dus:

$$\mathbf{X} = \mathbf{X} + \mathbf{v}t \tag{2}$$

Vul je $t = 0$ in, dan krijg je de positie in frame 1; vul je $t = 1$, dan krijg je de positie in frame 2 (geen rekening gehouden met het effect van een mogelijke botsing).

We moeten nu een vergelijking opzetten en oplossen voor t, waarin we de relatieve afstand tussen twee ballen gelijkstellen aan de som van de stralen:

$$\begin{aligned} \|\mathbf{X_1} + \mathbf{v_1}t - \mathbf{X_2} - \mathbf{v_2}t\| &= r_1 + r_2 \\ \|\mathbf{X_{rel}} + \mathbf{v_{rel}}t\| &= r_1 + r_2 \end{aligned} \tag{3}$$

Beide kanten kwadrateren geeft:

$$\begin{aligned} \mathbf{v_{rel}}^2 t^2 + 2\mathbf{X_{rel}}\mathbf{v_{rel}} + \mathbf{X_{rel}}^2 &= (r_1 + r_2)^2 \\ (\mathbf{v_{rel}} \cdot \mathbf{v_{rel}})t^2 + 2(\mathbf{X_{rel}} \cdot \mathbf{v_{rel}})t + \mathbf{X_{rel}} \cdot \mathbf{X_{rel}} &= (r_1 + r_2)^2 \\ (\mathbf{v_{rel}} \cdot \mathbf{v_{rel}})t^2 + 2(\mathbf{X_{rel}} \cdot \mathbf{v_{rel}})t + \mathbf{X_{rel}} \cdot \mathbf{X_{rel}} - (r_1 + r_2)^2 = 0 \end{aligned} \tag{4}$$

Deze vergelijking is een simpele ABC-formule, die ook in een iets alternatieve vorm kan worden geschreven om rekentijd te beperken:

$$\begin{aligned}
at^2 + 2bt + c &= 0 \\
t^2 + \tfrac{2b}{a}t &= -\frac{c}{a} \\
\left(t + \tfrac{b}{a}\right)^2 &= \frac{b^2}{a^2} - \frac{c}{a} \\
t + \tfrac{b}{a} &= \pm\sqrt{\frac{b^2 - ac}{a^2}} \\
t &= \frac{-b \pm \sqrt{b^2 - ac}}{a}
\end{aligned} \tag{5}$$

Vergelijking (5) kunnen we nu toepassen op vergelijking (4):

$$\begin{aligned}
a &= \mathbf{v_{rel}} \cdot \mathbf{v_{rel}} \\
b &= \mathbf{X_{rel}} \cdot \mathbf{v_{rel}} \\
c &= \mathbf{X_{rel}} \cdot \mathbf{X_{rel}} - (r_1 + r_2)^2
\end{aligned} \tag{6}$$

2.1.2 Simulatie

We hebben deze formule omgezet naar een simulatie in Geogebra, om te kijken of formule klopte; zie de bijlage met de bestandsnaam Bal-Bal.ggb. Zowel de verplaatsingsvectoren, beginposities en de stralen van beide cirkels zijn aan te passen, en als er binnen het begin- en eindpunt een botsing is, worden de cirkels opnieuw gestippeld getekend op de plaats van de botsing.

2.2 Cirkel tegen lijnstuk

Om erachter te komen of een cirkel een lijnstuk raakt binnen een tijdinterval, gebruiken we een formule die het punt op het lijnsegment geeft dat het dichtst bij het midden van de cirkel ligt. Voor de rest nog even hard nadenken hoe het gaat.

2.2.1 Punt op lijnstuk dichtste bij cirkel

Omdat we met vectoren werken is het heel makkelijk om te bepalen welk punt op een lijnstuk het dichtst bij het middelpunt van een cirkel ligt. Voor het gemak noemen we de uiteinden van het lijntuk A en B, het middelpunt van de cirkel C, en we definiëren P als het punt op het lijnstuk het dichtste bij C. P is dus een loodrechte projectie van C op AB. Verder stellen we het lijnstuk voor als een vector $\mathbf{v}$ vanuit punt A en tekenen we een vector $\mathbf{w}$ vanuit A naar C. Eerder bleek al dat geldt (1):

$$\begin{aligned} \mathbf{v} \cdot \mathbf{w} &= \|\mathbf{v}\| \cdot \|\mathbf{w}\| \cdot \cos\gamma \\ \cos\gamma &= \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \cdot \|\mathbf{w}\|} \end{aligned} \tag{7}$$

Verder geldt in driehoek $\triangle ACP$, waarbij $AC = \mathbf{w}$:

$$\cos\gamma = \frac{AP}{\|\mathbf{w}\|} = \frac{x}{\|\mathbf{w}\|} \tag{8}$$

We zijn nu op zoek naar de verhouding tussen x en $\mathbf{w}$. Dat geeft een indicatie van hoever punt P op het lijnstuk ligt. Is die verhouding kleiner dan 0, of groter dan 1, dan ligt het punt dat het dichtste bij C ligt, voorbij het lijnstuk. Ligt het ertussen, dan ligt het punt ergens tussen A en B. Combineer je (7) en (8), dan krijg je:

$$\begin{aligned} \frac{x}{\|\mathbf{w}\|} &= \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \cdot \|\mathbf{w}\|} \\ x &= \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\|} \\ \frac{x}{\|\mathbf{v}\|} &= \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \cdot \|\mathbf{v}\|} \\ \frac{x}{\|\mathbf{v}\|} &= \frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{v} \cdot \mathbf{v}} \end{aligned} \tag{9}$$

We hebben dit ook in Geogebra gesimuleerd, zie in de bijlage het bestand Bal-Lijn.ggb. Variabelen zijn hierbij het de positie van het de begin- en eindpunten van het lijnstuk (dat is meteen al vector $\mathbf{v}$ geworden) en de positie van de cirkel.

3 Gevolgen van botsingen

Hier schrijft Derk een lang verhaal.

4 Programmeren

Hier komt nog een verhaal, maar het resultaat is veel belangrijker.