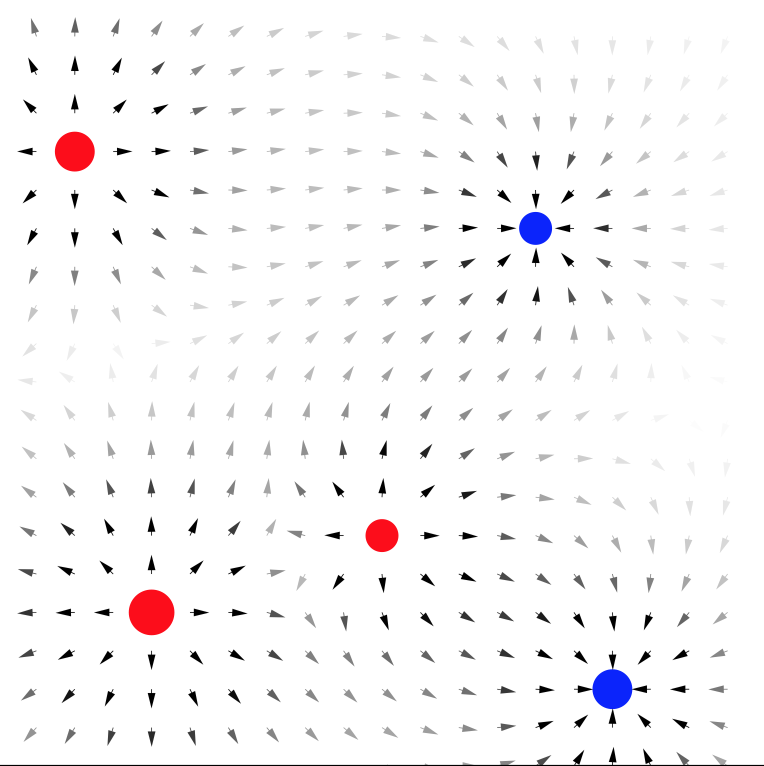
Graphische Darstellung von elektrischen Feldern

**Elektrische Felder beliebiger Ladungsverteilungen**

**Im Unterricht hast Du qualitativ einige elektrische Felder verschiedener Ladungskonfigurationen gesehen. Qualitativ deshalb, weil es von Hand sehr aufwendig ist, das elektrische Feld von Ladungskonfigurationen von mehr als einer Ladung zu berechnen. In diesem Modul sollst Du einen vorgefertigten Computercode so anpassen, dass er diese rechenintensive Aufgabe für (fast) jede beliebige Ladungskonfiguration berechnet.**

Du lernst in diesem Modul

* Dich in einem bestehenden Code zu orientieren
* eine Fragestellung zu vereinfachen, zu lösen und wieder zu verallgemeinern
* eine Modellierung auf Papier in Code zu übersetzen
* Antworten im Internet zu finden
* Wissen aus der Mathematik praktisch anzuwenden

Geh zuerst auf https://repl.it/@ThomasBisig/Electric-Field-Lines und lass den Code laufen ('Run' drücken). Wenn der Code fertig ausgeführt wurde, schau Dir das generierte Bild 'field\_vectors.png' an.

**A1.** Was könnte auf diesem Bild dargestellt sein? Welche Teile erkennst Du? Was kann nicht korrekt sein?

**1. Physikalische Grundlagen: Coulombgesetz**

Aus dem Physikunterricht wissen wir, welche Kraft zwischen zwei Ladungen Q1 und Q2 besteht.

**A2**. Such in der Formelsammlung nach dem **Coulombgesetz** und notiere Dir das Gesetz mit den zwei Ladungen Q1 und Q2 im Abstand r.

Wir haben im Unterricht gesehen, dass wir statt der Kraft einer Ladung Q1 auf eine zweite Ladung Q2 auch das sogenannte **elektrische Feld** einer (Quell)ladung Q betrachten können. Dieses elektrische Feld ergibt sich, wenn wir die Coulombkraft aufteilen. Sie gibt an, wie stark und in welche Richtung eine Probeladung q im Raum verschoben wird (hier haben wir stillschweigend Q2 durch q ausgetauscht – weil q eine sehr kleine Ladung sein soll).

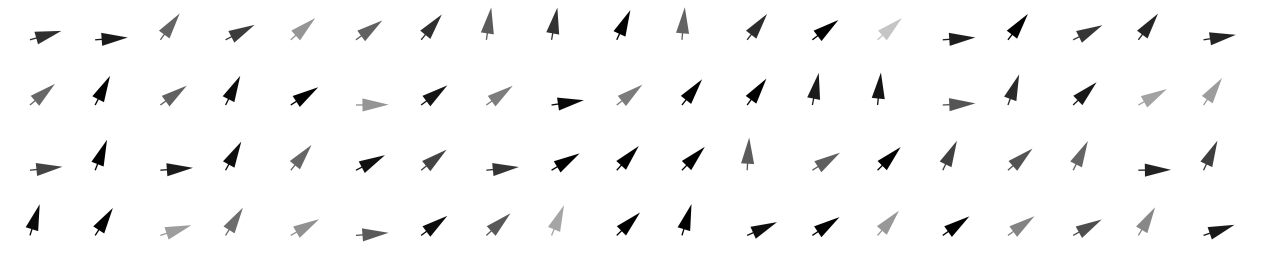
**A3**. Schau im Physikskript nach, wie wir das elektrische Feld definiert haben und notiere Dir die Herleitung.

Stellen wir uns den Raum den wir betrachten wollen als zweidimensionale Fläche vor, dann können wir den Raum als *xy*-Koordinatensystem abstrahieren (die dritte Dimension kannst Du als herausfordernde Aufgabe selber implementieren).

**Gelernt**

Du hast das Wissen über elektrische Felder und Kräfte repetiert und verstehst, wieso diese Konzepte zentral sind, um unser Visualisierungsproblem zu lösen.

**2. Feldvektoren in der Physik**



Elektrische Felder – wie auch Kräfte – werden als Vektoren beschrieben.

**A4**. Durch welche zwei Grössen ist ein Vektor bestimmt?

Die Stärke der Kraft wird normalerweise mit der Länge des Vektors dargestellt. In unserem Beispiel ist dies aber nicht sehr praktisch, da sich (so wie der Code programmiert ist) die Pfeile überschneiden würden.

**A5**. Anstelle der Länge des Vektors, was wurde zur Darstellung der Stärke im Titelbild verwendet?

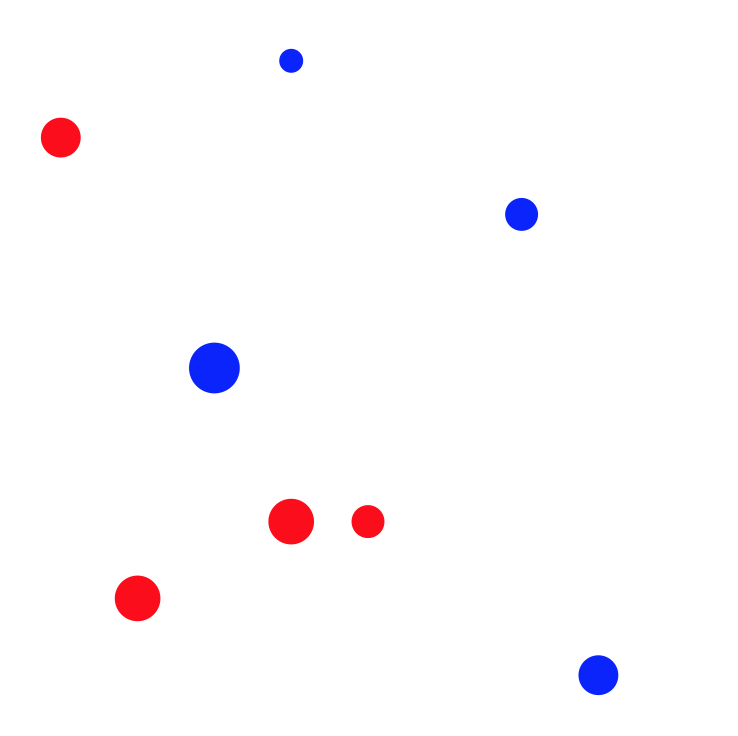
**A6.** Zeichne ein Koordinatensystem mit nur positiven x- und y-Achsen, beide Achsen skaliert von 0 – 10. Zeichne dann an einem beliebigen Punkt im Koordinatensystem eine Ladung positive Ladung ein (ein Punkt). Wähle zwei weitere Punkt und überlege Dir qualitativ, wie die Vektoren des elektrischen Feldes aussehen müssten. Zeichne diese zwei Vektoren ein.

**Gelernt**

Du hast Dein Wissen über Vektoren aufgefrischt und kannst dieses mathematische Konzept auf das Beispiel der Kräfte zwischen zwei Ladungen anwenden.

**3. Reduktion auf eine Ladung**

Im Titelbild siehst Du das elektrische Feld von unterschiedlichen Ladungen. Wie wir gesehen haben, wäre das von-Hand-rechnen für dieses Beispiel sehr aufwendig. Statt mit mehreren Ladungen, beginnen wir deshalb mit nur einer Ladung. Wenn wir diese Aufgabe gelöst haben (und überprüft haben, dass das Vektorfeld korrekt ist), können wir weitere Ladungen hinzufügen und uns überlegen, was wir anpassen müssen.



**A7**. Finde im Code die Zuweisung der Ladungen (Tipp: Englisch) und überlege Dir, wie die Du die Angaben (d.h. so wie die Ladungen gegeben sind) interpretieren könntest.

Tipp: Welche Dinge musst Du über die einzelne Ladung wissen? Lass Deine Antwort von der Lehrperson überprüfen.

**A8**. Verändere die Liste der Ladungen so, dass nur noch eine Ladung mit Stärke  
1e-9 C im Punkt P(0.4/0.4) bleibt. Lass das Programm laufen.

Hinweis: Wie Du gleich sehen wirst, ist im bestehenden Code der Faktor "e-9" ausgelagert in die Variable *charge\_scale*. D.h. Du musst Dich nur um die "ganzen Zahlen" kümmern.

**A9**. Wenn Du das Bild aus A8 anschaust: was hättest Du erwartet bzw. was ist anders als erwartet?

Die Richtung und Stärke der Pfeile werden (noch) zufällig generiert. D.h. dass an jedem Punkt, an welchem Du einen Pfeil siehst, nicht die richtige Berechnung des elektrischen Feldes ausgeführt wird, sondern zwei zufällige Grössen

field\_x = max\_field\*random.random()

filed\_y = max\_field\*random.random()

zugeordnet werden. *random.random()* generiert eine Zufallszahl und *max\_field* ist ein Skalierungsfaktor.[[1]](#footnote-1)

**A10**. Auf was könnte sich die Grösse field\_x und field\_y beziehen?

Du siehst, dass zwischen den Zeilen 53 und 63 einiges an Code fehlt. Deine Aufgabe ist es, diesen Teil (d.h. die Berechnung der Komponenten des elektrischen Feldes) zu programmieren.

Zuerst ist es hilfreich, eine Zeichnung zu machen und die einzelnen Punkte zu beschriften:

**A11**. Zeichne zum untenstehenden Bild Achsen und beschrifte folgende Dinge:

* die Ladung c (für charge) sei der ungefüllte Kreis,
* den Punkt im Raum, welcher uns interessiert, mit p (point),
* die Abstände in x- und y-Richtung (schau im Code nach wie sie beschrieben werden),
* den absoluten Abstand von c und p (im Code zu finden),
* sowie den E-Feld-Vektor f (wenn Q eine positive Ladung ist) in Komponenten zerlegt (die wiederum im Code zu finden sind).

Wenn Du nun zurückblätterst zu **A3**, kannst Du feststellen, welche Grössen wir – um die Formel anwenden zu können – berechnen müssen.

* Die Ladung(en) sind bereits vorgegeben => keine Berechnung
* ε0 und Pi können wir aus den Bibliotheken importieren (siehe später) => keine Berechnung
* Der Abstand r zwischen der Ladung und dem Ort, an welchem wir das Feld berechnen wollen, fehlt => Berechnung notwendig

Um den Abstand r zu berechnen gehen wir wie folgt vor:

**A12**. Übertrage – mit Hilfe der Grössen in c und p – die Abstände von der Quellladung c zum Ort des Interesses p in x und y Richtung in den Code. (Abstände werden immer von der Ladung weg gemessen).

Achtung: bei den Grössen c und p handelt sich um Listen (bzw. Arrays) für jedes c wie auch jedes p. Schau im Skript zum ersten Modul nach, was das heisst oder nutze die Suchmaschine Deiner Wahl um Dich zu informieren.

**A13**. Wie kannst Du nun den absoluten Abstand r mit Hilfe der zwei vorher definierten Grössen finden? Schau Dir die obige Skizze an und wende einen der fundamentalen mathematischen Sätze der Geometrie an. Was für eine Funktion brauchst Du, um das zu berechnen? Nutze das Internet, um die notwendige Hilfe zu erhalten.

**A14**. Mit Hilfe der Formel aus **A3** kannst Du nun die Berechnung für die Feldstärke f in den Code übernehmen.

Tipp: Du kannst Pi als Dezimalzahl eingeben oder die vorgespeicherte Grösse Pi aus der *math* Bibliothek benutzen: pi

Du kannst den Code laufen lassen, jedoch wird den zwei Komponenten field\_x und field\_y immer noch die zwei Zufallszahlen zugewiesen (Zeilen 65/66). Wir wollen nun – mit Hilfe der vorher berechneten Feldstärke f – die Komponenten berechnen, die Du in der Skizze von **A11** oben eingezeichnet hast.

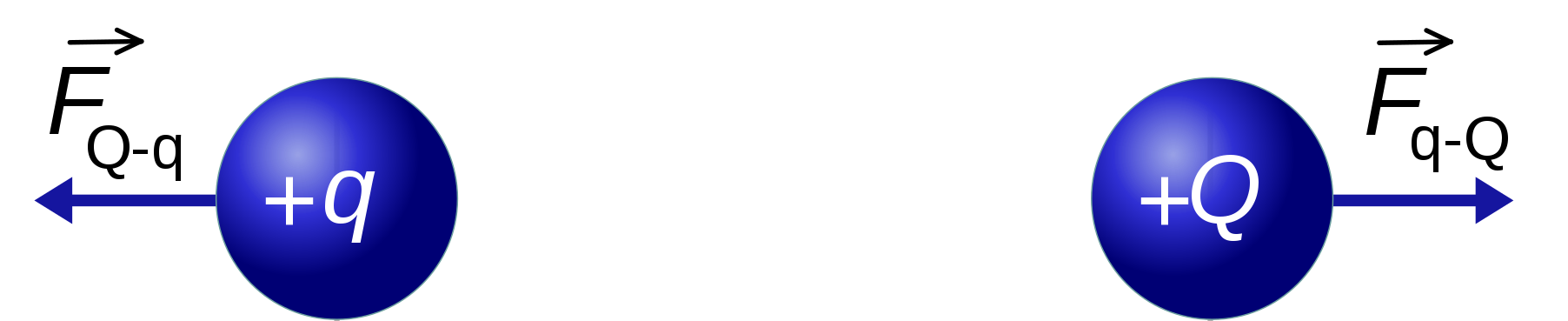
**A15**. Bevor Du im Skript weiterliest: diskutiere und finde mögliche Wege, die Komponenten field\_x und field\_y zu finden. Nutze dazu Deine Skizze von vorhin und Dein Wissen über Winkel(sätze). Lass Deine Ideen von der Lehrperson überprüfen.

**A16**. Falls Du nicht mindestens 5min selber versucht hast, eine Lösung zu **A15** zu finden, gehe nochmals zurück und nutze das Internet, um eine Lösung zu finden.

Wie können wir die zwei Komponenten (bzw. auch Vektoren) finden, die vektoraddiert unseren Feldstärkevektor erzeugen?

Angenommen wir hätten den Steigungswinkel des Dreiecks f, field\_x und field\_y (siehe Deine Skizze oben oder mach Dir eine neue), könnten wir mit Hilfe der Trigonometrie die Grössen berechnen. Aber wie finden wir den Winkel? In welche Richtung zeigt der Vektor?

Wenn wir zurück ans Coulombgesetz denken – mit zwei Ladungen Q und q – wird der elektrische Feldvektor in der Verbindungslinie von Q und q liegen. Damit liegt der elektrische Feldvektor auch auf der Verbindungsgerade von c und p. (In welche Richtung zeigt der Vektor f in Deiner Skizze in **A11**?)



**A17**. Welche zwei Grössen in Kombination mit welcher Winkelfunktion gibt somit den Winkel an, mit welchem der Feldvektor f von p weg/hinzeigt? Such im Internet nach der Funktion in Kombination mit *"Python Math Library"* und lass Dein Resultat durch die Lehrperson prüfen. Weise der Variablen *angle* auf Zeile 58 das Resultat der Winkelfunktion (= einen Winkel) zu.

**A18**. Gegeben sei der Winkel (wie oben berechnet) im Dreieck f, field\_x und field\_y, wie auch f. Wie findest Du nun die zwei Katheten field\_x und field\_y?

Tipp: Nutze weitere zwei Winkelsätze.

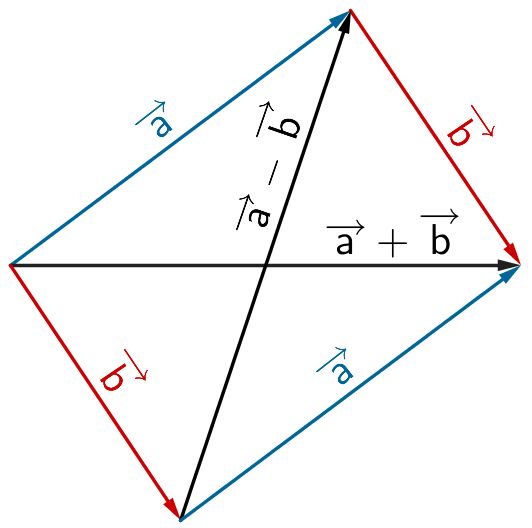
Schreibe alle hergeleiteten Berechnungen in den Code und kommentiere die Zeilen 65/66 aus (schreibe ein "#" zuforderst in die Zeile). Lass dann den Code laufen. Ergibt das Bild Sinn?

**A19**. Wenn Du eine weitere Ladung hinzufügst, was geschieht (oder eben nicht)?

**Gelernt**

Du hast Dein Wissen aus dem Physikunterricht über Vektoren, Kräfte und das Coulombgesetz angewendet. Du hast die Winkelfunktionen in einem rechtwinkligen Dreieck angewedet, um die Katheten bzw. die Winkel zu berechnen.

**3. Verallgemeinerung auf mehrere Ladungen**



**A20**. Es seien zwei Vektoren gegeben: a = (3/4) und b = (5/-3). Wie gross ist a + b?

Wie Du aus der Vektorgeometrie weisst, kann man Vektoren komponentenweise addieren. D.h. das 'Aneinanderhängen' von Vektoren (grafisch), wird rechnerisch über das Addieren der einzelnen Komponenten gelöst. Genau das wollen wir auch in unserem Beispiel anwenden.

Wenn mehrere Ladungen anwesend sind (passe die Ladungen gegebenenfalls an) wirken sich alle Ladungen auf das elektrische Feld in einem gewissen Punkt im Raum aus. D.h. wir müssen (für die Berechnung des "Nettofeldes") die Feldvektoren von allen Ladungen berechnen und diese addieren. Im Algorithmus heisst das, dass wir über alle Ladungen iterieren und ihren Feldvektor berechnen. Dann können wir komponentenweise addieren.

**A21**. Wie könntest Du das im Code umsetzen? (Die Iteration = Schlaufe/Loop ist bereits auf Zeile 49 gegeben).

Tipp: Wenn Du einer bestehenden Grösse x sich selber plus eine Zahl zuweisen willst (zum Beispiel pro Iteration die Zahl um eins erhöhen – zum Zählen) kann ich das folgendermassen machen: x += 1. Dies ist genau dasselbe wie x = x + 1.[[2]](#footnote-2)

**A22**. Generiere nun mindesten 5 Bilder mit unterschiedlichen Ladungskonfigura-tionen.

Fragen

* Funktioniert unser Code auch mit negativen Ladungen?
* Wie ist die Grösse der einzelnen Ladungen dargestellt? Findest Du den Ort im Code, wo die Zuweisung passiert?
* Herausforderung: Kannst Du das Beispiel so umprogrammieren, dass die Länge der Pfeile der Stärke der Pfeile entspricht?

1. Siehe Kapitel 3.4 im letzten Skript [↑](#footnote-ref-1)
2. Siehe A12 im letzten Skript. [↑](#footnote-ref-2)