DN2

Arbeitsblatt: DNET1

Name: **Kurznamen:**

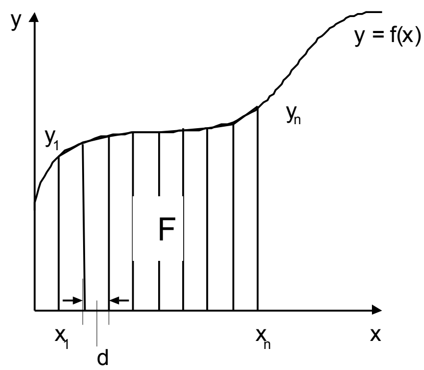
Bestimmung des zurückgelegten Weges

Zur Bestimmung der zurückgelegten Weglänge bei gegebenem Geschwindigkeitsverlauf muss in einem v-t Diagramm die Fläche unterhalb der v(t)-Kurve bestimmt werden. Ist die Geschwindigkeit konstant oder der Verlauf eine einfache Funktion der Zeit, dann kann die Weglänge mittels *Integration* analytisch berechnet werden. Ist hingegen der Geschwindigkeitsverlauf nur empirisch, d.h. durch Messung ermittelbar oder eine komplexe Funktion, so werden numerische Methoden angewandt.

Für die Bestimmung der Position eines Flugzeugs (oder Raumschiffs) wird im Trägheitnavigationssystem ein ähnliches Verfahren angewandt. Dabei wird die Beschleunigung in jede der 3-Achsenrichtungen gemessen (a-t Verlauf) und mittels Integration die aktuelle Geschwindigkeit (v-t Verlauf) ermittelt. Aus dieser lässt sich dann mit dem oben beschriebenen Verfahren die Position bestimmen.

Das Trapezverfahren zur numerischen Integration:

Beim Trapezverfahren wird die zu berechnende Fläche F unter einer Kurve y(x) zwischen dem Startwert x_1 und dem Endwert x_n in einzelne Trapeze der Höhe d unterteilt. Für jedes dieser Trapeze wird die Fläche berechnet und die Flächen werden anschliessend aufsummiert. Dies führt zu der untenstehenden Formel für die numerische Integration. Der Länge des d-Intervalls kann frei gewählt werden, z.b.$(x_n – x_1)$ / 100;



Fläche $F(x_1, x_n) = d*(y_1 + 2*y_2 + 2*y_3 + ... + 2*y_{n-1} + y_n)/2$

Aufgabe 1

Entwickeln Sie eine Klasse `Integrator` mit einer Methode `Integrate`, mit der Sie beliebige Funktionen numerisch über einen Bereich integrieren können.
Hinweise:

- Überlegen Sie sich genau, welche Stützwerte mit welchem Gewicht mitgezählt werden müssen.
- Unterteilen Sie den Bereich in eine vordefinierte Anzahl Schritte, z.B. 100
- Testen Sie Ihre Methode anhand einer einfachen Funktion, wie z.B. f(x) = x (sollte im Bereich 0..10 **exakt** 50 ergeben).
- Die Funktion ist als `Func<double, double> f` definiert. Dadurch kann die Funktion als Lambda Ausdruck, z.B. `x => x` beim Aufruf übergeben werden und innerhalb der Integrate Methode einfach mit `f(x)` aufgerufen werden. Wir werden Lambda Ausdrücke in einer späteren Vorlesung noch ausführlich behandeln; hier nur eine kleine Anwendung davon.
- Verwenden Sie das vorgegebene Gerüst.

Abgabe:

Praktikum: DN2.1

Filename: Integrator.cs

Aufgabe 2 Adaptives Verfahren

Statt einer konstanten Anzahl Schritte kann auch bis zu einer vorgegebenen Genauigkeit gerechnet werden (glatte Funktionen kommen mit wesentlich weniger Stützwerten aus als stark oszillierende).

Hinweis: Die Genauigkeit bzw. Grösse des Fehlers schätzen Sie durch die Änderungen des Integralwertes ab, der sich durch eine *Verdoppelung der Stützwerte* ergibt. Falls dieser eine vorgegebene Grenze (z.B. 0.001) unterschreitet, brechen Sie die Berechnung ab.

Hinweis:

- Korrekte Ausgabe am Schluss

```
Linear fixed [0..10]: 50 steps: 100
Linear fixed [5..15]: 100 steps: 100
Linear adapt [0..10]: 50 steps: 2
Square fixed [0..10]: 333.35 steps: 100
Square adapt [0..10]: 333.33333581686 steps: 8192
```

Abgabe:

Praktikum: DN2.2

Filename: Integrator.cs