

■ **Wir bestimmen die Flugzeit in der Ebene**

Zeit[{ax, ay}, {bx, by}, {gwx, gwy}, gf]

bestimmt die Flugzeit in der Ebene zwischen den Punkten

{ax,ay} und {bx,by}

wobei die Windgeschwindigkeit als Vektor

{gwx,gwy}

beträgt und die Fluggeschwindigkeit

gf

ist.

```
Zeit[{ax_, ay_}, {bx_, by_}, {gwx_, gwy_}, gf_] := Module[
  {gAB, g, W, Lsg, C, C1, C2, AC, AB, BC1, BC2, t1, t2, t},
  t = ∞;

  gAB = {ax, ay} + g * {bx - ax, by - ay};
  W = {ax + gwx, ay + gwy};
  Lsg = Solve[(gAB - W) . (gAB - W) == gf^2, g];
  C1 = gAB /. Lsg[[1]];
  C2 = gAB /. Lsg[[2]];
  BC1 = C1 - {bx, by};
  BC2 = C2 - {bx, by};
  AB = {bx - ax, by - ay};

  t1 = Solve[AB == tvar * (C1 - {ax, ay}), tvar][[1, 1, 2]];
  t2 = Solve[AB == tvar * (C2 - {ax, ay}), tvar][[1, 1, 2]];

  If[t1 > 0 && t2 > 0, t = Min[t1, t2],
  If[t1 > 0 && t2 ≤ 0, t = t1,
  If[t1 ≤ 0 && t2 > 0, t = t2,
  Print["keine Lösung"]
  ]
  ]
];

t
]
```

Unser erstes kleines Beispiel:

Zeit[{0, 0}, {100, 400}, {0, 300}, 800] \* 60 // N

22.7434

Wenn wir über mehrere Punkte fliegen wollen, dann benötigen wir einen Algorithmus.

Wir navigieren entlang der Punkte in Liste1:

Liste1 = {{0, 0}, {25, 100}, {50, 200}, {75, 300}, {100, 400}}

{{0, 0}, {25, 100}, {50, 200}, {75, 300}, {100, 400}}

ZeitGesamt[Liste, Vektorfeld, gf]

bestimmt die Flugzeit in der Ebene entlang der Punkte in

Liste

wobei die Windgeschwindigkeit in

Vektorfeld

definiert ist und die Fluggeschwindigkeit

gf

ist.

```

ZeitGesamt[Liste_, Vektorfeld_, gf_] := Module[
  {t, L, i, ti, A, B},
  t = 0;
  L = Length[Liste];

  For[i = 1, i ≤ L - 1, i++,
    A = Liste[[i]];
    B = Liste[[i + 1]];
    ti = Zeit[A, B, (Vektorfeld[A] + Vektorfeld[B]) / 2, gf];
    t = t + ti;
  ];

  t
]

```

Unser erstes "kleines" Vektorfeld:

```
Vektorfeld1[{x_, y_}] := {0, 300}
```

Damit erhalten wir folgende Flugzeit:

```
ZeitGesamt[Liste1, Vektorfeld1, 800] * 60 // N
```

```
22.7434
```

#### ■ Wir modellieren den Wind

Den Wind modellieren wir über Messpunkte, die wir mit B - Splines interpolieren.

Dazu benötigen wir Knoten in x-Richtung

```
knotsX = {-180, 180};
```

und Knoten in y-Richtung

```

knotsY = {-90, -90, -70,
  -50,
  -35,
  -20,
  0,
  20,
  35,
  50,
  70, 90, 90};

```

Außerdem benötigen wir Basisfunktionen für die x- und y-Richtung:

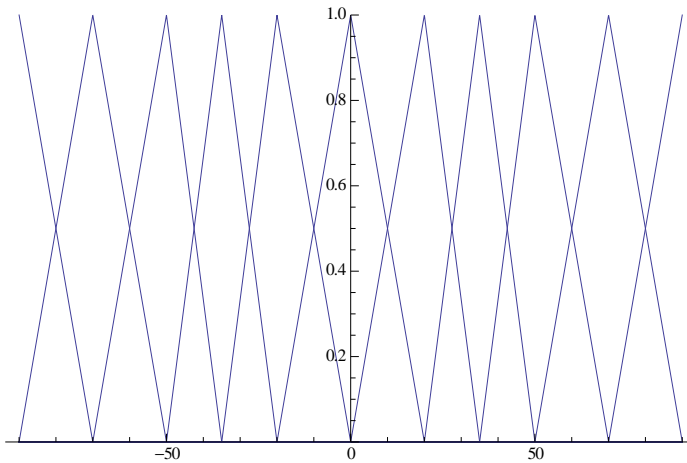
```

BasisX[i_, x_] := BSplineBasis[{0, knotsX}, i, x]
BasisY[i_, y_] := BSplineBasis[{1, knotsY}, i, y]

```

Die Basisfunktionen für die y-Richtung sehen so aus:

```
Plot[
  Table[
    BasisY[i, y],
    {i, 0, 10}],
  {y, -90, 90}, PlotRange -> {0, 1}]
```



Jetzt können wir den Wind an bestimmten Messpunkten angeben:

```
WindMessungen = {
  {
    {0, 0},
    {25, 0},
    {500, 0},
    {50, 0},
    {200, 0},
    {0, 0},
    {500, 0},
    {50, 0},
    {500, 0},
    {25, 0},
    {0, 0}
  }
};
```

Daraus ergibt sich der Wind als Vektorfeld, dass die Windmesspunkte interpoliert:

```
Wind[{y_, x_}] := Sum[WindMessungen[[i, j]] * BasisX[i - 1, x] * BasisY[j - 1, y], {i, 1, 11}, {j, 1, 11}]
```

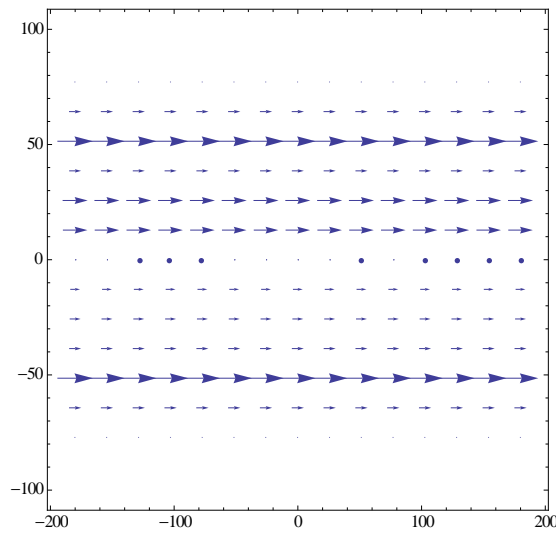
Eine kleine Messung:

```
Wind[{-50, 0}] // N
```

```
{500., 0.}
```

und ein Bild unseres Vektorfelds:

```
VectorPlot[Wind[{y, x}], {x, -180, 180}, {y, -90, 90}]
```



#### ■ Wir zeichnen die Welt

Dazu benötigen wir das Package WorldPlot

```
<< WorldPlot`
```

Jetzt können wir uns eine Weltkarte zeichnen:

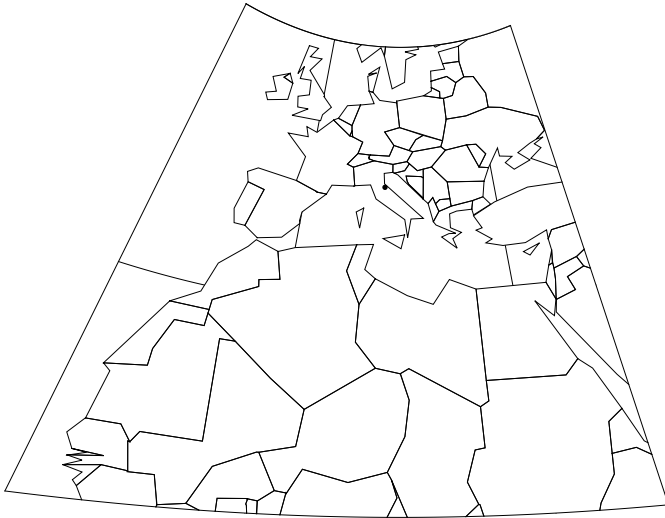
```
WorldPlot[World]
```

Wir definieren eine eigene Abbildung der Kugel:

```
Abbildung[b_, l_] :=
{
  Cos[b] * Sin[l - 10] /
  (Sin[b1] * Sin[b] + Cos[b1] * Cos[b] * Cos[l - 10]),
  (Cos[b1] * Sin[b] - Sin[b1] * Cos[b] * Cos[l - 10]) /
  (Sin[b1] * Sin[b] + Cos[b1] * Cos[b] * Cos[l - 10])
} /. {l0 -> Pi / 12, b1 -> Pi / 4}
```

... und erkennen, dass wir sie nicht vernünftig verwenden können:

```
WorldPlot[World, WorldProjection -> (Abbildung[#1 * Pi / (60 * 180.), #2 * Pi / (60 * 180.)] &),
WorldRange -> {{10, 60}, {-20, 40}}]
```



#### ■ Wir bestimmen Entfernungen auf der Kugel

Die Entfernung zwischen zwei Orten (in Länge und Breite gegeben) auf der Kugel:

```
Entf[{b1_, l1_}, {b2_, l2_}] := ArcCos[Sin[b1 * Pi / 180] * Sin[b2 * Pi / 180] +
Cos[b1 * Pi / 180] * Cos[b2 * Pi / 180] * Cos[(l2 - l1) * Pi / 180]] * 6371
```

```
Wien = {48 + 12 / 60, 16 + 22 / 60};
```

```
Kefermarkt = {48 + 27 / 60, 14 + 32 / 60};
```

```
Honolulu = {21 + 19 / 60, -(157 + 50 / 60)};
```

```
NewYork = {40 + 43 / 60, -74};
```

```
Sydney = {-(33 + 51 / 60), 151 + 12 / 60};
```

```
Peking = {39 + 56 / 60, 116 + 23 / 60};
```

von Kefermarkt nach Honolulu in km:

```
Entf[Kefermarkt, Honolulu] // N
```

```
12 220.2
```

von Kefermarkt nach Wien in km:

```
Entf[Kefermarkt, Wien] // N
```

```
138.363
```

Die Zeit, die man auf der Kugel benötigt (so ungefähr):

```

ZeitGlobus[A_, E_, Wind_, gf_] :=
Module[{H1, H2, a, b, c, d, DD, BB, wx, wy, wxH, wyH, Ausgabe},
  H1 = {A[[1]], E[[2]]};
  H2 = {E[[1]], A[[2]]};
  a = Entf[A, H2] // N;
  b = Entf[H2, E] // N;
  c = Entf[E, H1] // N;
  d = Entf[H1, A] // N;
  DD = {
    (c - a) / 2,
    Sqrt[b^2 - (c - a)^2 / 4]
  };
  BB = {
    c,
    0
  };
  {wx, wy} = Wind[(A + E) / 2];

  If[{A[[1]] ≥ E[[1]]} && {A[[2]] < E[[2]]},
    {wxH, wyH} = {wx, wy};
  ];
  If[{A[[1]] ≥ E[[1]]} && {A[[2]] ≥ E[[2]]},
    {wxH, wyH} = {-wx, wy};
  ];
  If[{A[[1]] < E[[1]]} && {A[[2]] < E[[2]]},
    {wxH, wyH} = {wx, -wy};
  ];
  If[{A[[1]] < E[[1]]} && {A[[2]] ≥ E[[2]]},
    {wxH, wyH} = {-wx, -wy};
  ];

  If[Abs[E[[2]] - A[[2]]] ≤ 360 - Abs[E[[2]] - A[[2]]],
    Ausgabe = Zeit[DD, BB, {wxH, wyH}, gf],
    Ausgabe = Zeit[DD, BB, {-wxH, wyH}, gf]
  ];

  Ausgabe
]

```

Der Wind über Kefermarkt:

```
Wind[Kefermarkt] // N
```

```
{453.5, 0.}
```

von Kefermarkt nach Wien:

```
ZeitGlobus[Kefermarkt, Wien, Wind, 800] * 60 // N
```

```
10.9799
```

knapp unter 7 Minuten.

## ■ Wir konstruieren einen Großkreis mit Bisektion

```

Mittelpunkt[{b1_, l1_}, {b2_, l2_}] := Module[
  {lm, bm, ax, bx, ay, by, az, bz, mx, my, mz, Betragm, m0x, m0y, m0z, bmB, lmB},
  lm = 0;
  bm = 0;
  ax = Cos[l1 * Pi / 180] * Cos[b1 * Pi / 180];
  ay = Sin[l1 * Pi / 180] * Cos[b1 * Pi / 180];
  az = Sin[b1 * Pi / 180];
  bx = Cos[l2 * Pi / 180] * Cos[b2 * Pi / 180];
  by = Sin[l2 * Pi / 180] * Cos[b2 * Pi / 180];
  bz = Sin[b2 * Pi / 180];
  mx = ax + bx;
  my = ay + by;
  mz = az + bz;
  Betragm = Sqrt[mx^2 + my^2 + mz^2];
  m0x = mx / Betragm;
  m0y = my / Betragm;
  m0z = mz / Betragm;
  bmB = ArcSin[m0z];
  bm = bmB * 180 / Pi;

  If[(m0x > 0 && m0y > 0) || (m0x > 0 && m0y < 0), lm = ArcTan[m0y / m0x] * 180 / Pi];
  If[(m0x < 0 && m0y > 0), lm = ArcTan[m0y / m0x] * 180 / Pi + 180];
  If[(m0x < 0 && m0y < 0), lm = ArcTan[m0y / m0x] * 180 / Pi - 180];
  If[(m0x < 0 && m0y == 0), lm = -180];
  If[(m0x > 0 && m0y == 0), lm = 0];
  If[(m0x == 0 && m0y > 0), lm = 90];
  If[(m0x == 0 && m0y < 0), lm = -90];

  {bm, lm} // N
]

Halbieren[Liste_] :=
Module[
  {Ausgabe, n},
  n = Length[Liste];

  Ausgabe = Table[
    If[EvenQ[i],
      Mittelpunkt[Liste[[i/2]], Liste[[i/2 + 1]]
    ,
      Liste[[i/2]]
    ]
    , {i, 1, n + n - 1}];

  Ausgabe
]

Halbieren[Halbieren[{Kefermarkt, Honolulu}]]

```

$$\left\{ \left\{ \frac{969}{20}, \frac{218}{15} \right\}, \{75.3109, 0.679721\}, \{75.4787, -140.02\}, \{48.6273, -154.083\}, \left\{ \frac{1279}{60}, -\frac{947}{6} \right\} \right\}$$

```

Teilen[Liste_, n_] := Module[
  {Ausgabe, i},

  Ausgabe = Liste;

  For[i = 1, i ≤ n, i++,
    Ausgabe = Halbieren[Ausgabe];
  ];

  Ausgabe
]

Show[{WorldPlot[World, WorldRotation → {90, 0, 0}, WorldRange → {{0, 90}, {-180, 180}},
  WorldProjection → LambertAzimuthal], WorldGraphics[{Red, Thickness[0.003], Line[
  Teilen[{Kefermarkt, NewYork}, 10] * 60
  ]}]}]

```



```

ZeitGlobusGesamt[Liste_, Wind_, gf_] :=
Module[
  {n, i, ZeitGes},

  ZeitGes = 0;
  n = Length[Liste];
  For[i = 1, i ≤ n - 1, i++,
    ZeitGes = ZeitGes + ZeitGlobus[Liste[[i]], Liste[[i + 1]], Wind, gf];
  ];
  ZeitGes
]

KeinWind[{x_, y_}] := {0, 0}

```

■ Wir definieren besseren Wind

```

knotsX = {-180, -180, -120, -60, 0, 60, 120, 180, 180};
knotsY = {-90, -90, -80,
  -40,

```



```

-35,
-20,
0,
20,
30,
45,
75, 90, 90};
BasisX[i_, x_] := BSplineBasis[{1, knotsX}, i, x]
BasisY[i_, y_] := BSplineBasis[{1, knotsY}, i, y]
WindMessungen = {
  {
    {0, 0},
    {25, -3},
    {350, 80},
    {50, 50},
    {200, 80},
    {0, 0},
    {500, 0},
    {100, 0},
    {500, 0},
    {25, 0},
    {0, 0}
  }, {
    {0, 0},
    {25, -2},
    {350, 180},
    {50, 10},
    {200, 90},
    {0, 0},
    {450, 80},
    {100, 70},
    {400, -100},
    {25, 0},
    {0, 0}
  },
  {
    {0, 0},
    {25, 0},
    {500, 0},
    {150, 0},
    {200, 0},
    {0, 0},
    {480, 50},
    {100, 0},
    {300, 0},
    {25, 0},
    {0, 0}
  },
  {
    {0, 0},
    {25, 0},
    {450, -90},
    {50, 0},
    {200, -80},
    {0, 0},
    {400, 80},

```

```

      {80, 0},
      {450, 0},
      {25, 0},
      {0, 0}
    },
    {
      {0, 0},
      {25, 0},
      {500, 0},
      {50, 0},
      {200, 0},
      {0, 0},
      {500, 90},
      {100, 0},
      {500, 15},
      {25, 0},
      {0, 0}
    },
    {
      {0, 0},
      {25, 0},
      {500, 0},
      {50, 0},
      {200, 0},
      {0, 0},
      {500, 0},
      {100, 0},
      {500, -100},
      {25, 0},
      {0, 0}
    },
    {
      {0, 0},
      {25, -3},
      {350, 80},
      {50, 50},
      {200, 80},
      {0, 0},
      {500, 0},
      {100, 0},
      {500, 0},
      {25, 0},
      {0, 0}
    }
  };

```

```
Wind[{y_, x_}] := Sum[WindMessungen[[i, j]] * BasisX[i - 1, x] * BasisY[j - 1, y], {i, 1, 7}, {j, 1, 11}]
```

#### ■ Wir bestimmen die Flugzeit

```
ZeitGlobusGesamt[Teilen[{Peking, Honolulu}, 10], KeinWind, 850]
```

```
9.60132
```

```
ZeitGlobusGesamt[Teilen[{Honolulu, Peking}, 10], KeinWind, 850]
```

```
9.5999
```

```
ZeitGlobusGesamt[Teilen[{Peking, Honolulu}, 10], Wind, 850]
```

```
9.54301
```

```
ZeitGlobusGesamt[Teilen[{Honolulu, Peking}, 10], Wind, 850]
```

```
11.5924
```

```
MinTabelle[Funkt_, n_, step_, arg1_, arg2_] :=
Module[
  {fWert, i, j, fMin, arg1Min, arg2Min},

  fMin = Infinity;
  arg1Min = arg1;
  arg2Min = arg2;

  For[i = -n, i ≤ n, i++,
    For[j = -n, j ≤ n, j++,
      fWert = Funkt[arg1 + step*i, arg2 + step*j];
      If[fWert < fMin, fMin = fWert; arg1Min = arg1 + step*i; arg2Min = arg2 + step*j];
    ];
  ];

  {fMin, {arg1Min, arg2Min}}
]
```

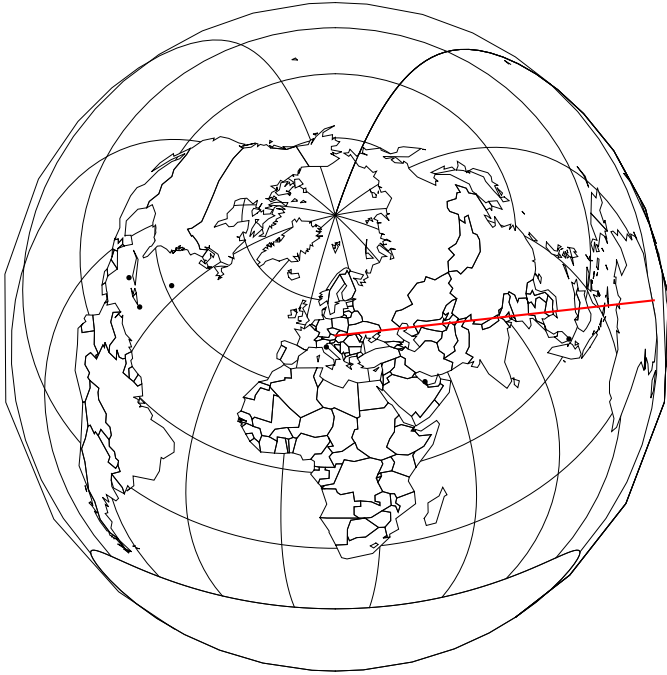
```
Start = Wien;
```

```
Ziel = Peking;
```

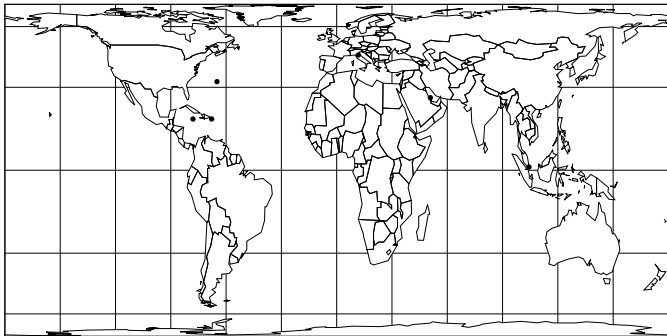
```
Show[{WorldPlot[World], WorldGraphics[{Red, Thickness[0.003], Line[
  Teilen[{Start, Ziel}, 10] * 60
]}]}]
```



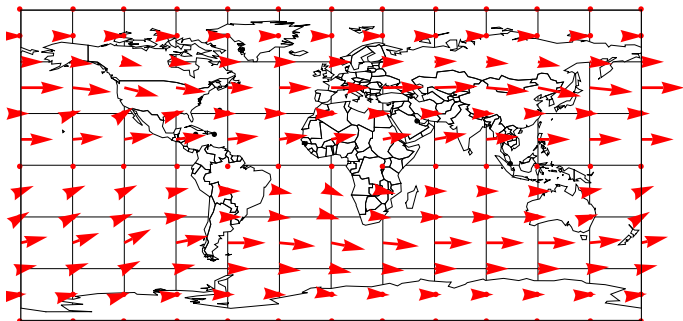
```
Show[{WorldPlot[World,  
  WorldRotation -> {Start[[1]], 0, -Start[[2]]},  
  WorldRange -> {{-60, 90}, {-180, 180}}, WorldProjection -> LambertAzimuthal},  
WorldGraphics[{Red, Thickness[0.003], Line[  
  Teilen[{Start, Ziel}, 10] * 60  
  ]}}]
```



```
Show[{WorldPlot[World, WorldProjection -> LambertCylindrical]]}
```



```
Show[
  {WorldPlot[World], WorldGraphics[Table[Table[{Arrowheads[Medium], Red, Thickness[0.004], Arrow[
    {{x, y}*60, {x, y}*60 + Wind[{y, x]}*3}
  ]
  }, {x, -180, 180, 30}], {y, -90, 90, 15}]]]}]
```



```
ZeitGlobusGesamt[Teilen[{Start, Ziel}, 10], KeinWind, 850]
```

18.795

```
ZeitGlobusGesamt[Teilen[{Start, Ziel}, 10], Wind, 850]
```

17.4105

```
Flugzeit[Liste_] := N[ZeitGlobusGesamt[Teilen[Liste, 4], Wind, 850]]
```

```
{b1m1, l1m1} = Mittelpunkt[Start, Ziel];
{b1m0, l1m0} = Mittelpunkt[Start, {b1m1, l1m1}];
{b1m2, l1m2} = Mittelpunkt[{b1m1, l1m1}, Ziel];
```

```
Funkt[b1_, l1_] := Flugzeit[{Start, {b1m0, l1m0}, {b1, l1}, {b1m2, l1m2}, Ziel]}
{fmin, {b1m1min, l1m1min}} = MinTabelle[Funkt, 4, 8, b1m1, l1m1]
```

```
Funkt[b1_, l1_] := Flugzeit[{Start, {b1, l1}, {b1m1min, l1m1min}, {b1m2, l1m2}, Ziel]}
{fmin, {b1m0min, l1m0min}} = MinTabelle[Funkt, 4, 8, b1m0, l1m0]
```

```
Funkt[b1_, l1_] := Flugzeit[{Start, {b1m0min, l1m0min}, {b1m1min, l1m1min}, {b1, l1}, Ziel]}
{fmin, {b1m2min, l1m2min}} = MinTabelle[Funkt, 4, 8, b1m2, l1m2]
```

{17.4309, {25.5885, 90.5366}}

{16.815, {56.3002, 42.2251}}

{16.8018, {6.94793, 106.918}}

```
{b1m1, l1m1} = {b1m1min, l1m1min};
{b1m0, l1m0} = {b1m0min, l1m0min};
{b1m2, l1m2} = {b1m2min, l1m2min};
```

```

Funkt[b1_, l1_] := Flugzeit[{Start, {blm0, l1m0}, {b1, l1}, {blm2, l1m2}, Ziel}]
{fmin, {blm1min, l1m1min}} = MinTabelle[Funkt, 3, 4, blm1, l1m1]

Funkt[b1_, l1_] := Flugzeit[{Start, {b1, l1}, {blm1min, l1m1min}, {blm2, l1m2}, Ziel}]
{fmin, {blm0min, l1m0min}} = MinTabelle[Funkt, 3, 4, blm0, l1m0]

Funkt[b1_, l1_] := Flugzeit[{Start, {blm0min, l1m0min}, {blm1min, l1m1min}, {b1, l1}, Ziel}]
{fmin, {blm2min, l1m2min}} = MinTabelle[Funkt, 3, 4, blm2, l1m2]

{24.049, {-20., 4.}}
{23.8768, {48.5, 0.}}
{23.7492, {-8.5, -4.}}

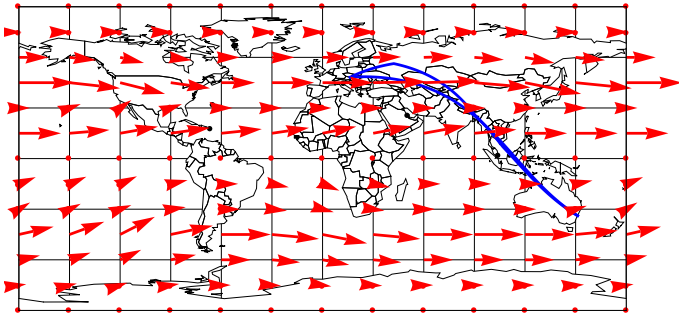
Show[{WorldPlot[World, WorldRange -> {{-90, 90}, {-180, 180}}],

WorldGraphics[{Blue, Thickness[0.004], Line[
  Teilen[{Start, Ziel}, 10] * 60
  ]}],

WorldGraphics[{Blue, Thickness[0.004], Line[
  Teilen[{Start, {blm0min, l1m0min}, {blm1min, l1m1min}, {blm2min, l1m2min}, Ziel}, 10] * 60
  ]}],

WorldGraphics[Table[Table[{Arrowheads[Medium], Red, Thickness[0.004], Arrow[
  {{x, y} * 60, {x, y} * 60 + Wind[{y, x}] * 4}
  ],
  ], {x, -180, 180, 30}], {y, -90, 90, 15}]]]

```



```
Show[{WorldPlot[World,  
  WorldRotation -> {Start[[1]], 0, -Start[[2]]},  
  WorldRange -> {{-80, 90}, {-180, 180}}, WorldProjection -> LambertAzimuthal],  
  
WorldGraphics[{Blue, Thickness[0.004], Line[  
  Teilen[{Start, Ziel}, 10] * 60  
  ]}],  
  
WorldGraphics[{Blue, Thickness[0.004], Line[  
  Teilen[{Start, {b1m0min, l1m0min}, {b1m1min, l1m1min}, {b1m2min, l1m2min}, Ziel}, 10] * 60  
  ]}]]]
```

