FODO lattice with error

In this case since there is an error in the lattice, the lattice has a superperiod of 1. Therefore, *all* elements have to be involved in calculating the beta function.

Transport matrices

In[•]:=
```
MQF[K_, l_] := ( Cos[Sqrt[K] l]                  1/Sqrt[K] Sin[Sqrt[K] l] )
                ( -Sqrt[K] Sin[Sqrt[K] l]        Cos[Sqrt[K] l]           );
MQD[K_, l_] := ( Cosh[Sqrt[K] l]                 1/Sqrt[K] Sinh[Sqrt[K] l] )
                ( Sqrt[K] Sinh[Sqrt[K] l]        Cosh[Sqrt[K] l]           );
Mdrift[l_] := ( 1  l )
              ( 0  1 );
```

Twiss matrices

$$\begin{pmatrix}\beta\\ \alpha\\ \gamma\end{pmatrix}_{s_2} = \begin{pmatrix} C^2 & -2SC & S^2 \\ -CC' & SC'+S'C & -SS' \\ C'^2 & -2S'C' & S'^2 \end{pmatrix} * \begin{pmatrix}\beta\\ \alpha\\ \gamma\end{pmatrix}_{s_1}.$$

In[•]:=
```
( Cqf   Sqf )
( Cpqf  Spqf ) = MQF[K, l]
```

Out[•]= $\{\{\mathrm{Cos}[\sqrt{K}\ l], \frac{\mathrm{Sin}[\sqrt{K}\ l]}{\sqrt{K}}\}, \{-\sqrt{K}\ \mathrm{Sin}[\sqrt{K}\ l], \mathrm{Cos}[\sqrt{K}\ l]\}\}$

In[•]:=
```
TQF[KK_, ll_] := ( Cqf^2          -2 Sqf Cqf          Sqf^2 )
                 ( -Cqf Cpqf      Sqf Cpqf + Spqf Cqf  -Sqf Spqf )
                 ( Cpqf^2         -2 Spqf Cpqf         Spqf^2 ) /. {K → KK, l → ll}
```

In[•]:=
```
( Cqd   Sqd )
( Cpqd  Spqd ) = MQD[K, l]
```

Out[•]= $\{\{\mathrm{Cosh}[\sqrt{K}\ l], \frac{\mathrm{Sinh}[\sqrt{K}\ l]}{\sqrt{K}}\}, \{\sqrt{K}\ \mathrm{Sinh}[\sqrt{K}\ l], \mathrm{Cosh}[\sqrt{K}\ l]\}\}$

In[•]:=
```
TQD[KK_, ll_] := ( Cqd^2          -2 Sqd Cqd          Sqd^2 )
                 ( -Cqd Cpqd      Sqd Cpqd + Spqd Cqd  -Sqd Spqd )
                 ( Cpqd^2         -2 Spqd Cpqd         Spqd^2 ) /. {K → KK, l → ll}
```

In[•]:=
$$\begin{pmatrix} Cl & Sl \\ Cpl & Spl \end{pmatrix} = \text{Mdrift}[l]$$

Out[•]= {{1, l}, {0, 1}}

In[•]:=
$$\text{Tdrift}[ll_] := \begin{pmatrix} Cl^2 & -2\,Sl\,Cl & Sl^2 \\ -Cl\,Cpl & Sl\,Cpl + Spl\,Cl & -Sl\,Spl \\ Cpl^2 & -2\,Spl\,Cpl & Spl^2 \end{pmatrix} /.\ l \to ll$$

Lattice

Let the number of FODO elements be three, i.e. I have

FODO|FODO| FODO
#1 | #2 | #3

Let the error be in FODO #2.

The error is ϵ

In[•]:=
```
Telement = {
  {TQF, k, lq/2}, {Tdrift, ld}, {TQD, k, lq}, {Tdrift, ld}, {TQF, k, lq/2},
  {TQF, k, lq/2}, {Tdrift, ld}, {TQD, k (1 + ϵ), lq}, {Tdrift, ld}, {TQF, k, lq/2},
  {TQF, k, lq/2}, {Tdrift, ld}, {TQD, k, lq}, {Tdrift, ld}, {TQF, k, lq/2}
 }
```

Out[•]=
$$\left\{\left\{TQF, k, \frac{lq}{2}\right\}, \{Tdrift, ld\}, \{TQD, k, lq\}, \{Tdrift, ld\}, \left\{TQF, k, \frac{lq}{2}\right\}, \left\{TQF, k, \frac{lq}{2}\right\}, \{Tdrift, ld\}, \{TQD, k\,(1+\epsilon), lq\}, \{Tdrift, ld\}, \left\{TQF, k, \frac{lq}{2}\right\}, \left\{TQF, k, \frac{lq}{2}\right\}, \{Tdrift, ld\}, \{TQD, k, lq\}, \{Tdrift, ld\}, \left\{TQF, k, \frac{lq}{2}\right\}\right\}$$

In[•]:=
```
Melement = {
  {MQF, k, lq/2}, {Mdrift, ld}, {MQD, k, lq}, {Mdrift, ld}, {MQF, k, lq/2},
  {MQF, k, lq/2}, {Mdrift, ld}, {MQD, k (1 + ϵ), lq}, {Mdrift, ld}, {MQF, k, lq/2},
  {MQF, k, lq/2}, {Mdrift, ld}, {MQD, k, lq}, {Mdrift, ld}, {MQF, k, lq/2}
 }
```

Out[•]=
$$\left\{\left\{MQF, k, \frac{lq}{2}\right\}, \{Mdrift, ld\}, \{MQD, k, lq\}, \{Mdrift, ld\}, \left\{MQF, k, \frac{lq}{2}\right\}, \left\{MQF, k, \frac{lq}{2}\right\}, \{Mdrift, ld\}, \{MQD, k\,(1+\epsilon), lq\}, \{Mdrift, ld\}, \left\{MQF, k, \frac{lq}{2}\right\}, \left\{MQF, k, \frac{lq}{2}\right\}, \{Mdrift, ld\}, \{MQD, k, lq\}, \{Mdrift, ld\}, \left\{MQF, k, \frac{lq}{2}\right\}\right\}$$

Nominal values

```
In[•]:= k = 0.54102;
        lq = 0.5;
        ld = 2.5;

In[•]:= βx[ s_, ϵ1_] :=
          Module[{MFODO1, MFODO2, MFODO3, lattice, ϕ, βs, αs, γs, xs, d0, d1, Tw},
           (* Make the lattice to find xs*)
           MFODO1 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
           MFODO2 = MQF[k, lq/2].Mdrift[ld].
               MQD[k * (1 + ϵ), lq].Mdrift[ld].MQF[k, lq/2] /. ϵ → ϵ1 // N;
           MFODO3 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
           lattice = MFODO1.MFODO2.MFODO3;

           (*got the lattice, so solve for the initial Twiss parameters, xs*)
           ϕ = ArcCos[ 1/2 Tr[lattice]] ;
           βs = lattice[[1, 2]] / Sin[ϕ] // N;
           αs = Chop[ (lattice[[1, 1]] - Cos[ϕ]) / Sin[ϕ] ] // N;
           γs = - lattice[[2, 1]] / Sin[ϕ] // N;

           xs = (βs
                 αs
                 γs);

           (*Finally, I can do the transport*)

           d0 = d1 = 0;
           Tw = IdentityMatrix[3];

           For[i = 1, i ≤ Length[Telement], i++,
            (*Is it a quad?*)
            If[(Telement[[i, 1]] == TQF) || (Telement[[i, 1]] == TQD),
             d1 += Telement[[i, 3]]; (*get length*)
             If[d0 ≤ s ≤ d1,
              Return[Telement[[i, 1]][Telement[[i, 2]], s - d0].Tw.xs /. ϵ → ϵ1];
             ];
             (*not in the quad, so transport through*)
             Tw = Telement[[i, 1]][Telement[[i, 2]], Telement[[i, 3]]].Tw /. ϵ → ϵ1;
            ];
```

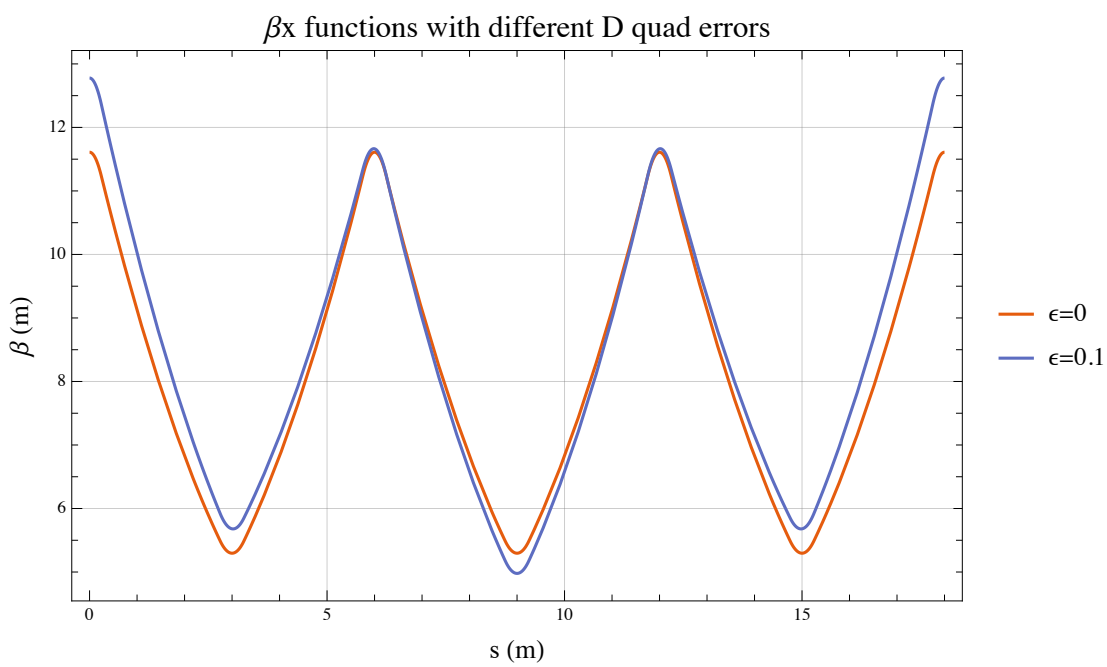
```
    (*Is it a drift?*)
    If[Telement⟦i, 1⟧ == Tdrift,
     d1 += Telement⟦i, 2⟧; (*get length*)
     If[d0 ≤ s ≤ d1,
      Return[Telement⟦i, 1⟧[s - d0].Tw.xs];
     ];
     (*not in the drift space, so transport through*)
     Tw = Telement⟦i, 1⟧[Telement⟦i, 2⟧].Tw;
    ];

    d0 = d1;
   ]; (*for*)
  ];
```

Plots

In[•]:=
```
g1 = Plot[{βx[s, 0]⟦1, 1⟧, βx[s, 0.1]⟦1, 1⟧} ,
  {s, 0, 3 * (lq / 2 + ld + lq + ld + lq / 2)}, PlotTheme → "Scientific",
  GridLines → Automatic, FrameLabel → {Style["s (m)", 14], Style["β (m)", 14]},
  PlotLabel → Style["βx functions with different D quad errors", 16],
  PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500]
```

Out[•]=



In[•]:=
```
Export[dir <> "lattice.pdf", g1]
```

StringJoin: String expected at position 1 in dir <> lattice.pdf.

Export: First argument dir <> lattice.pdf is not a valid file specification.

Out[•]=
```
$Failed
```

In[•]:=
```
Export[dir <> "betax_ideal.dat",
  Table[{s, βx[s, 0][[1, 1]]}, {s, 0, 3 * (lq / 2 + ld + lq + ld + lq / 2), 0.1}]];
```

StringJoin: String expected at position 1 in dir <> betax_ideal.dat.

Export: First argument dir <> betax_ideal.dat is not a valid file specification.

In[•]:=
```
Export[dir <> "betax_err.dat",
  Table[{s, βx[s, 0.1][[1, 1]]}, {s, 0, 3 * (lq / 2 + ld + lq + ld + lq / 2), 0.1}]];
```

StringJoin: String expected at position 1 in dir <> betax_err.dat.

Export: First argument dir <> betax_err.dat is not a valid file specification.

TBT data

I will assume that there is a BPM at the centre of each quad. Therefore, there are 6 BPMs.

The injection starts from the centre of FQ1

Centre of FQ1

In[•]:=
```
BPM1[x0_, n_, ϵ1_] := Module[{MFODO1, MFODO2, MFODO3, lattice, L, i, x},
   MFODO1 = MQF[k, lq / 2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq / 2] // N;
   MFODO2 = MQF[k, lq / 2].Mdrift[ld].
       MQD[k * (1 + ϵ), lq].Mdrift[ld].MQF[k, lq / 2] /. ϵ → ϵ1 // N;
   MFODO3 = MQF[k, lq / 2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq / 2] // N;
   lattice = MFODO3.MFODO2.MFODO1;

   L = {x0};

   x = x0;
   For[i = 1, i ≤ n, i++,
    x = lattice.x;
    L = Join[L, {x}];
   ];
   Return[L];
  ];
```

Centre of DQ2

```
In[•]:= BPM2[x0_, n_, ϵ1_] := Module[{MFODO1a, MFODO1b, MFODO2, MFODO3, lattice, L, i, x},
    MFODO1a = MQD[k, lq/2].Mdrift[ld].MQF[k, lq/2] // N;
    MFODO1b = MQF[k, lq/2].Mdrift[ld].MQD[k, lq/2] // N;
    MFODO2 = MQF[k, lq/2].Mdrift[ld].
        MQD[k * (1 + ϵ), lq].Mdrift[ld].MQF[k, lq/2] /. ϵ → ϵ1 // N;
    MFODO3 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    lattice = MFODO1a.MFODO3.MFODO2.MFODO1b;

    x = MFODO1a.x0;
    (*transport from injection to the centre of the quad*)
    L = {x};

    For[i = 1, i ≤ n, i++,
     x = lattice.x;
     L = Join[L, {x}];
    ];
    Return[L];
   ];
```

Centre of FQ3

```
In[•]:= BPM3[x0_, n_, ϵ1_] := Module[{MFODO1, MFODO2, MFODO3, lattice, L, i, x},
    MFODO1 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    MFODO2 = MQF[k, lq/2].Mdrift[ld].
        MQD[k * (1 + ϵ), lq].Mdrift[ld].MQF[k, lq/2] /. ϵ → ϵ1 // N;
    MFODO3 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    lattice = MFODO1.MFODO3.MFODO2;

    x = MFODO1.x0;
    (*transport from injection to the centre of the quad*)
    L = {x};


    For[i = 1, i ≤ n, i++,
     x = lattice.x;
     L = Join[L, {x}];
    ];
    Return[L];
   ];
```

Centre of DQ4

```
In[•]:= BPM4[x0_, n_, ϵ1_] := Module[{MFODO1, MFODO2a, MFODO2b, MFODO3, lattice, L, i, x},
    MFODO1 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    MFODO2a = MQD[k * (1 + ϵ), lq/2].Mdrift[ld].MQF[k, lq/2] /. ϵ → ϵ1 // N;
    MFODO2b = MQF[k, lq/2].Mdrift[ld].MQD[k * (1 + ϵ), lq/2] /. ϵ → ϵ1 // N;
    MFODO3 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    lattice = MFODO2a.MFODO1.MFODO3.MFODO2b;

    x = MFODO2a.MFODO1.x0;
    (*transport from injection to the centre of the quad*)
    L = {x};

    For[i = 1, i ≤ n, i++,
     x = lattice.x;
     L = Join[L, {x}];
    ];
    Return[L];
   ];
```

Centre of FQ5

```
In[•]:= BPM5[x0_, n_, ϵ1_] := Module[{MFODO1, MFODO2, MFODO3, lattice, L, i, x},
    MFODO1 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    MFODO2 = MQF[k, lq/2].Mdrift[ld].
        MQD[k * (1 + ϵ), lq].Mdrift[ld].MQF[k, lq/2] /. ϵ → ϵ1 // N;
    MFODO3 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    lattice = MFODO2.MFODO1.MFODO3;

    x = MFODO2.MFODO1.x0;
    (*transport from injection to the centre of the quad*)
    L = {x};

    For[i = 1, i ≤ n, i++,
     x = lattice.x;
     L = Join[L, {x}];
    ];
    Return[L];
   ];
```

Centre of DQ6

```
In[•]:= BPM6[x0_, n_, ϵ1_] := Module[{MFODO1, MFODO2, MFODO3a, MFODO3b, lattice, L, i, x},
    MFODO1 = MQF[k, lq/2].Mdrift[ld].MQD[k, lq].Mdrift[ld].MQF[k, lq/2] // N;
    MFODO2 = MQF[k, lq/2].Mdrift[ld].
        MQD[k * (1 + ϵ), lq].Mdrift[ld].MQF[k, lq/2] /. ϵ → ϵ1 // N;
    MFODO3a = MQD[k, lq/2].Mdrift[ld].MQF[k, lq/2] // N;
    MFODO3b = MQF[k, lq/2].Mdrift[ld].MQD[k, lq/2];
    lattice = MFODO3a.MFODO2.MFODO1.MFODO3b;

    x = MFODO3a.MFODO2.MFODO1.x0;
    (*transport from injection to the centre of the quad*)
    L = {x};

    For[i = 1, i ≤ n, i++,
     x = lattice.x;
     L = Join[L, {x}];
    ];
    Return[L];
   ];
```

Plots

```
In[•]:= dir = "~/expt/booster/lattice/math/";
```

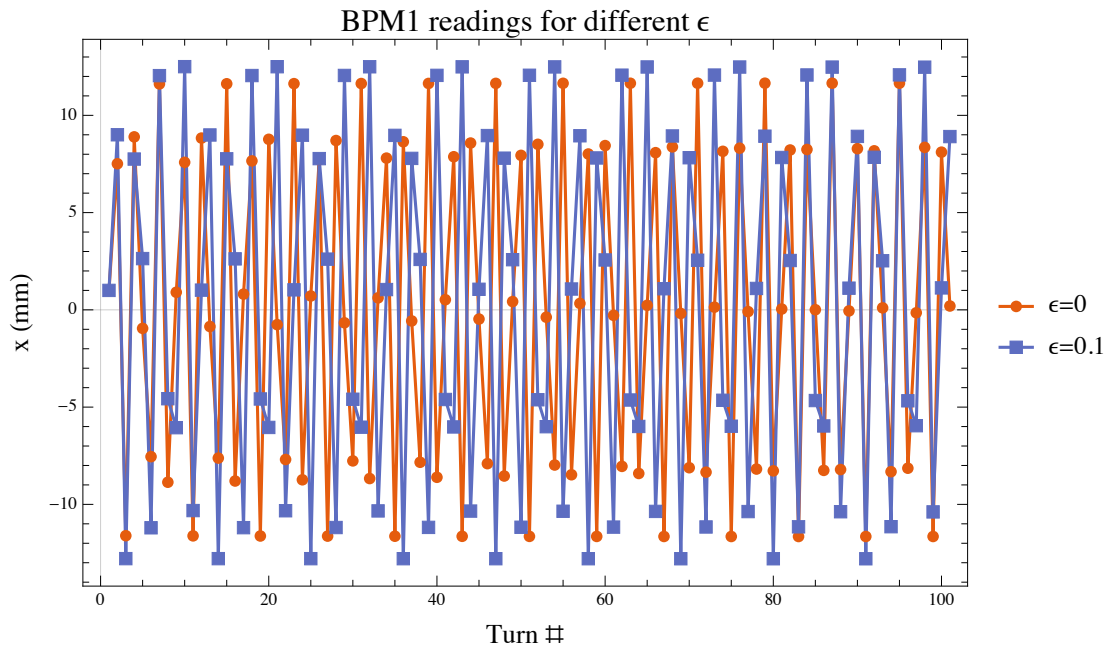
```
In[•]:= x0 = (0.001
              0.001);
```

```
In[•]:= ϵ0 = 0.1;
```

BPM1

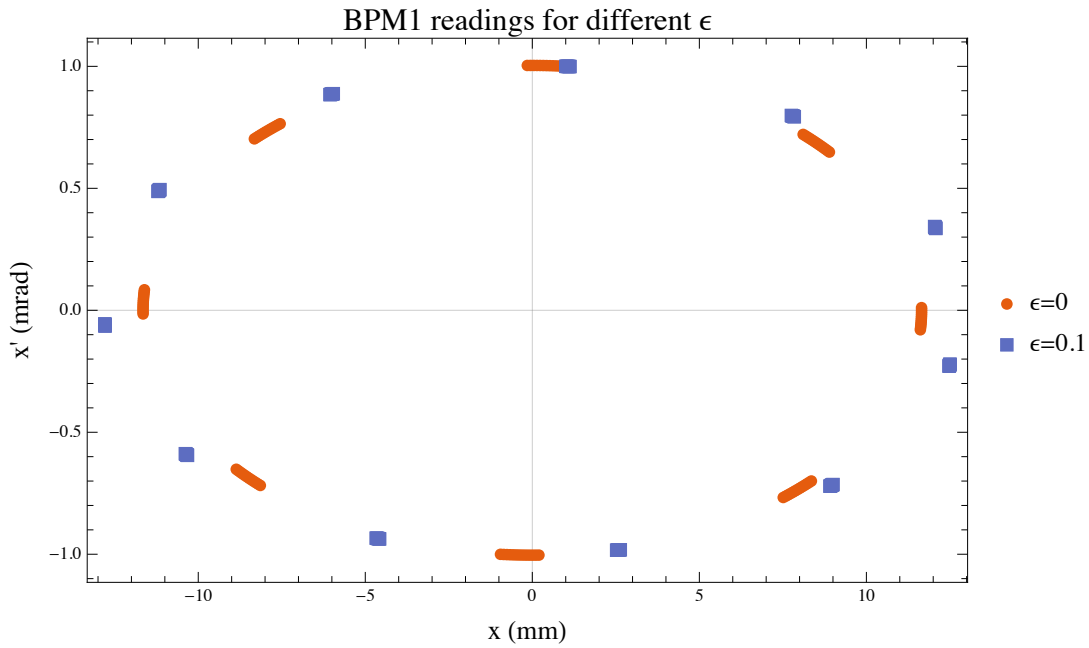
In[•]:=
```
g1 = ListPlot[{#[[1, 1]] 10^3 & /@ BPM1[x0, 100, 0], #[[1, 1]] 10^3 & /@ BPM1[x0, 100, ϵ0]},
  PlotMarkers → Automatic, Joined → True, PlotTheme → "Scientific",
  FrameLabel → {Style["Turn #", 14], Style["x (mm)", 14]},
  PlotLabel → Style["BPM1 readings for different ϵ", 16],
  PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
g2 = ListPlot[{(#ᵀ 10^3)[[1]] & /@ BPM1[x0, 100, 0], (#ᵀ 10^3)[[1]] & /@ BPM1[x0, 100, ϵ0]},
  PlotMarkers → Automatic, PlotTheme → "Scientific",
  FrameLabel → {Style["x (mm)", 14], Style["x' (mrad)", 14]},
  PlotLabel → Style["BPM1 readings for different ϵ", 16],
  PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



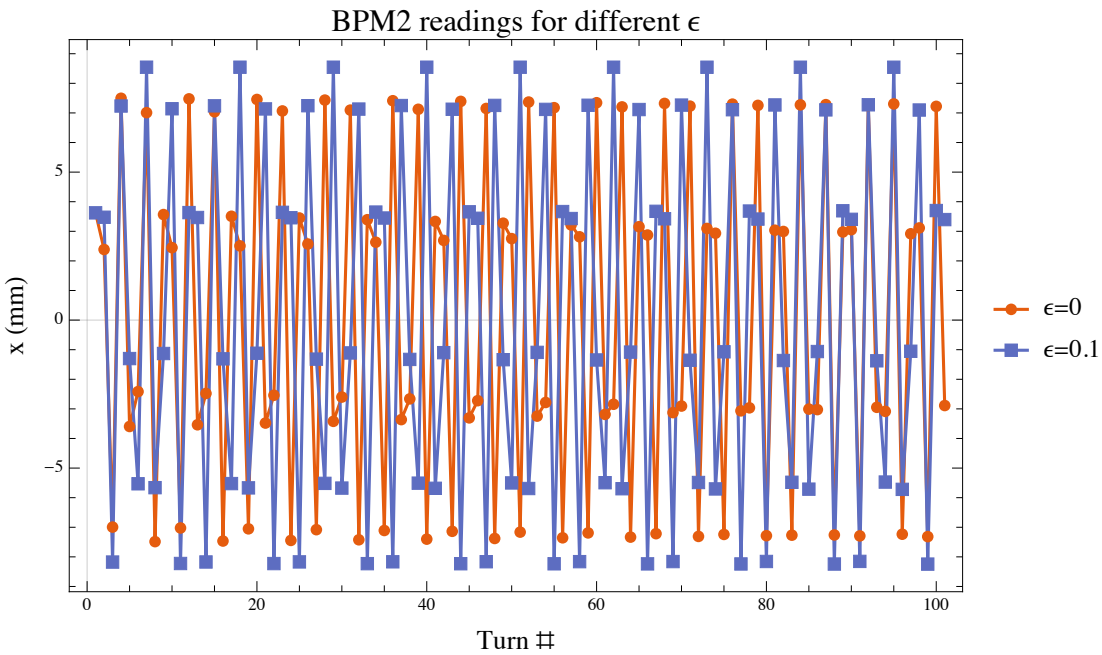
In[•]:=
```
Export[dir <> "bpm1_ideal.dat", (#ᵀ 10^3)[[1]] & /@ BPM1[x0, 100, 0]];
Export[dir <> "bpm1_err.dat", (#ᵀ 10^3)[[1]] & /@ BPM1[x0, 100, ϵ0]];
```

In[•]:=
```
Export[dir <> "bpm1pos.pdf", g1];
Export[dir <> "bpm1phasespace.pdf", g2];
```

BPM2

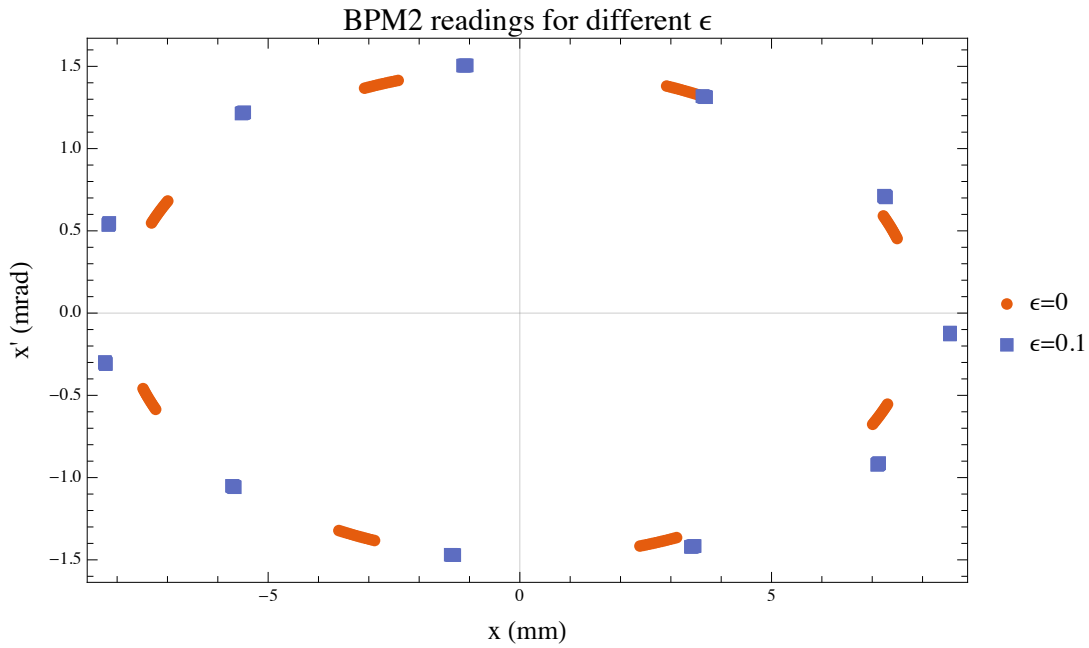
In[•]:=
```
ListPlot[{#[[1, 1]] 10^3 & /@ BPM2[x0, 100, 0], #[[1, 1]] 10^3 & /@ BPM2[(0.001, 0.001), 100, ϵ0]},
 PlotMarkers → Automatic, Joined → True, PlotTheme → "Scientific",
 FrameLabel → {Style["Turn #", 14], Style["x (mm)", 14]},
 PlotLabel → Style["BPM2 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
ListPlot[{(#ᵀ 10^3)[[1]] & /@ BPM2[x0, 100, 0], (#ᵀ 10^3)[[1]] & /@ BPM2[x0, 100, ϵ0]},
 PlotMarkers → Automatic, PlotTheme → "Scientific",
 FrameLabel → {Style["x (mm)", 14], Style["x' (mrad)", 14]},
 PlotLabel → Style["BPM2 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=

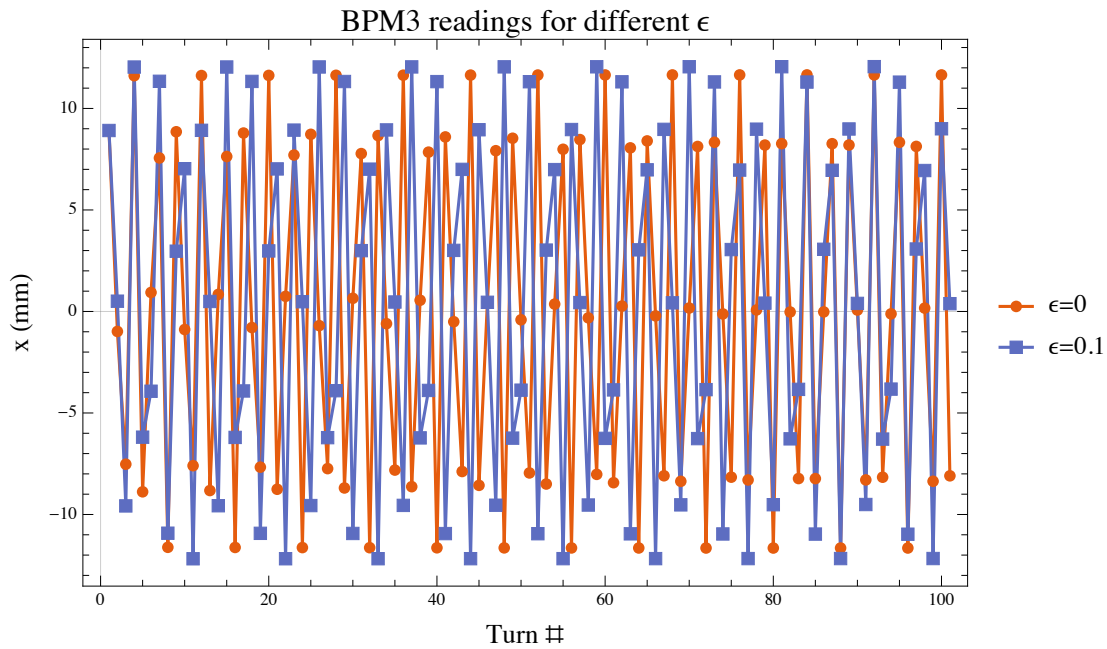


In[•]:=
```
Export[dir <> "bpm2_ideal.dat", (#ᵀ 10^3)[[1]] & /@ BPM2[x0, 100, 0]];
Export[dir <> "bpm2_err.dat", (#ᵀ 10^3)[[1]] & /@ BPM2[x0, 100, ϵ0]];
```

BPM3

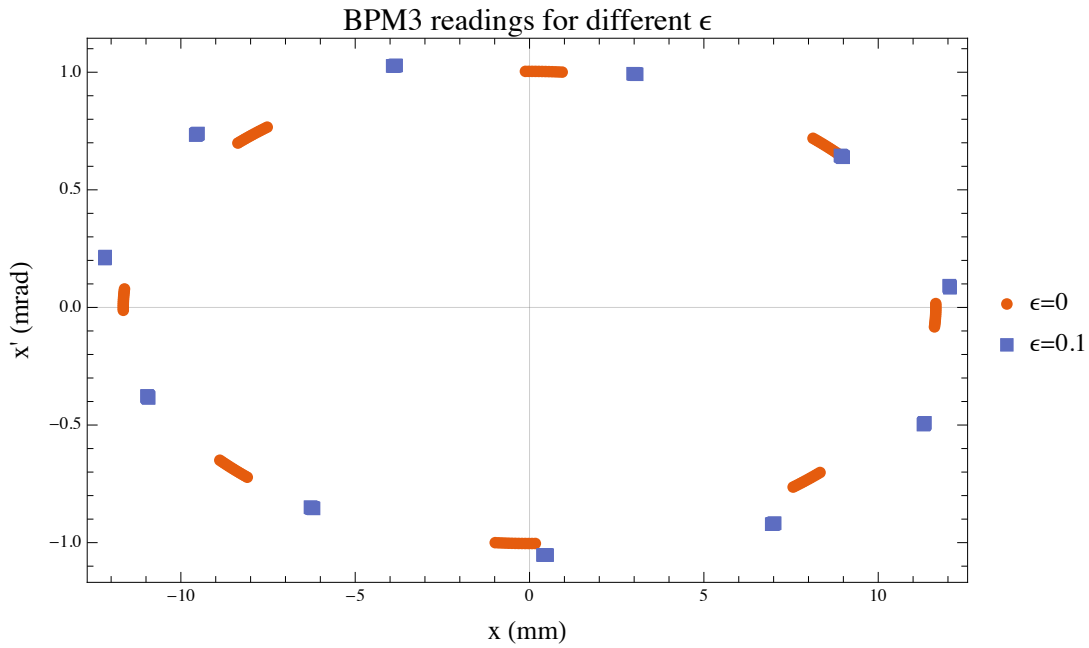
In[•]:=
```
ListPlot[{#[[1, 1]] 10^3 & /@ BPM3[x0, 100, 0], #[[1, 1]] 10^3 & /@ BPM3[x0, 100, ϵ0]},
 PlotMarkers → Automatic, Joined → True, PlotTheme → "Scientific",
 FrameLabel → {Style["Turn #", 14], Style["x (mm)", 14]},
 PlotLabel → Style["BPM3 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
ListPlot[{(#ᵀ 10^3)[[1]] & /@ BPM3[x0, 100, 0], (#ᵀ 10^3)[[1]] & /@ BPM3[x0, 100, ϵ0]},
 PlotMarkers → Automatic, PlotTheme → "Scientific",
 FrameLabel → {Style["x (mm)", 14], Style["x' (mrad)", 14]},
 PlotLabel → Style["BPM3 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=

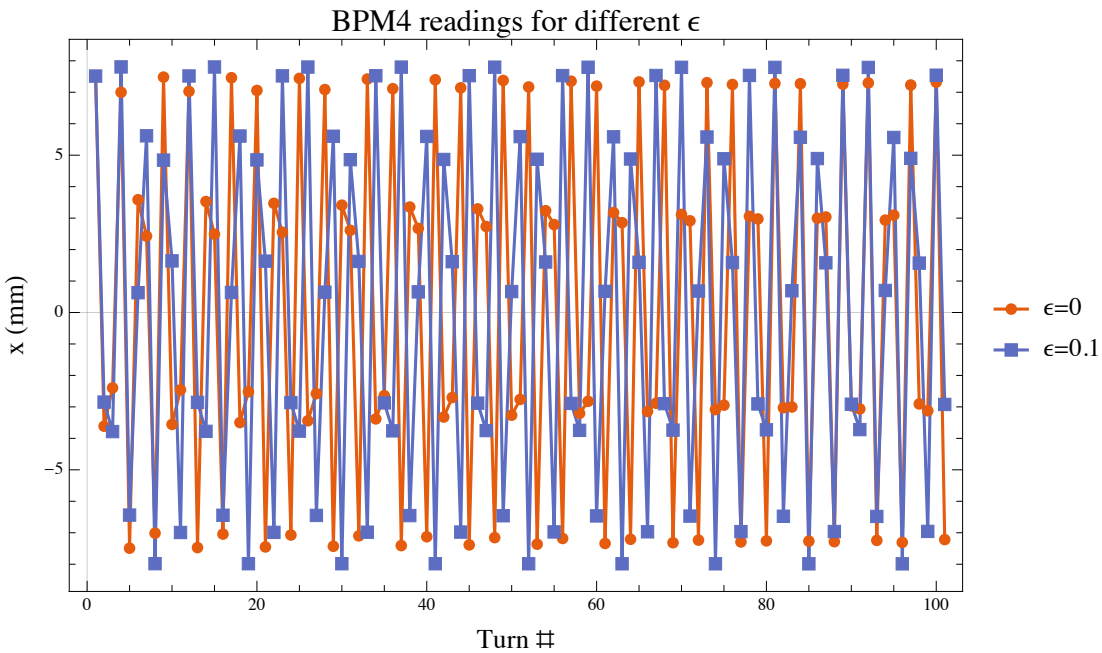


In[•]:=
```
Export[dir <> "bpm3_ideal.dat", (#ᵀ 10^3)[[1]] & /@ BPM3[x0, 100, 0]];
Export[dir <> "bpm3_err.dat", (#ᵀ 10^3)[[1]] & /@ BPM3[x0, 100, ϵ0]];
```

BPM4

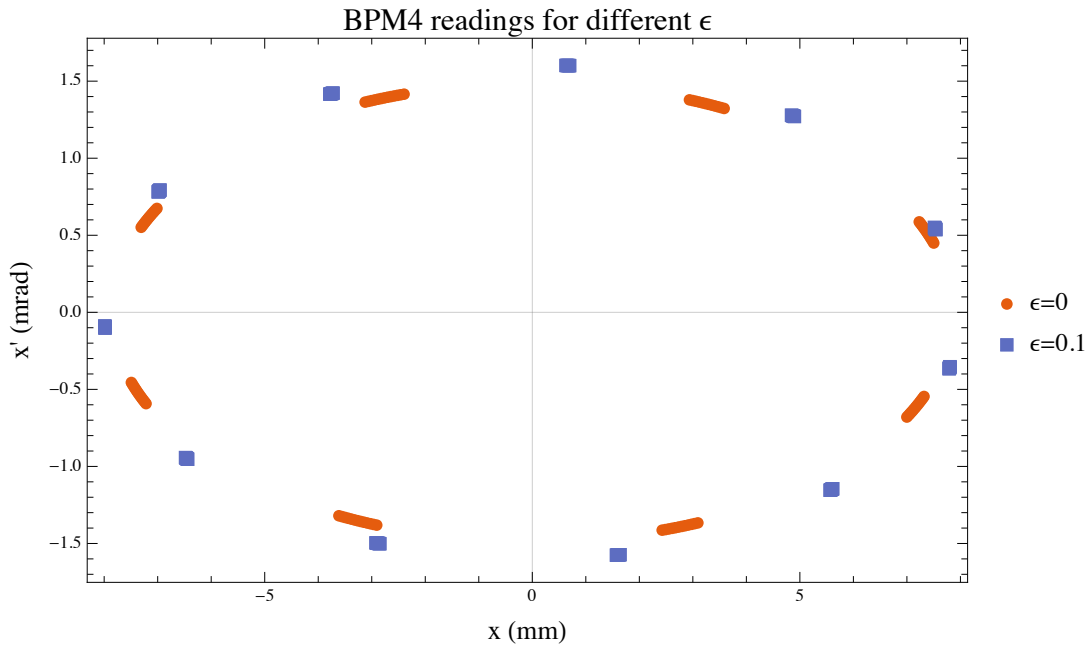
In[•]:=
```
ListPlot[{#[[1, 1]] 10^3 & /@ BPM4[x0, 100, 0], #[[1, 1]] 10^3 & /@ BPM4[x0, 100, ϵ0]},
 PlotMarkers → Automatic, Joined → True, PlotTheme → "Scientific",
 FrameLabel → {Style["Turn #", 14], Style["x (mm)", 14]},
 PlotLabel → Style["BPM4 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
ListPlot[{(#ᵀ 10^3)[[1]] & /@ BPM4[x0, 100, 0], (#ᵀ 10^3)[[1]] & /@ BPM4[x0, 100, ϵ0]},
 PlotMarkers → Automatic, PlotTheme → "Scientific",
 FrameLabel → {Style["x (mm)", 14], Style["x' (mrad)", 14]},
 PlotLabel → Style["BPM4 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=

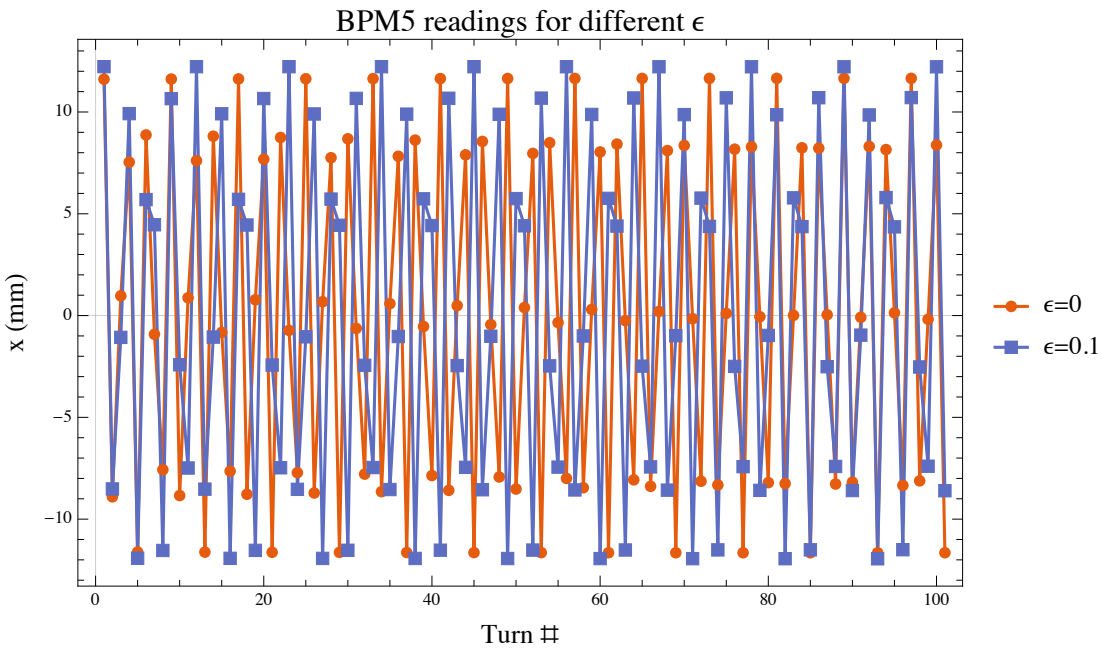


In[•]:=
```
Export[dir <> "bpm4_ideal.dat", (#ᵀ 10^3)[[1]] & /@ BPM4[x0, 100, 0]];
Export[dir <> "bpm4_err.dat", (#ᵀ 10^3)[[1]] & /@ BPM4[x0, 100, ϵ0]];
```

BPM5

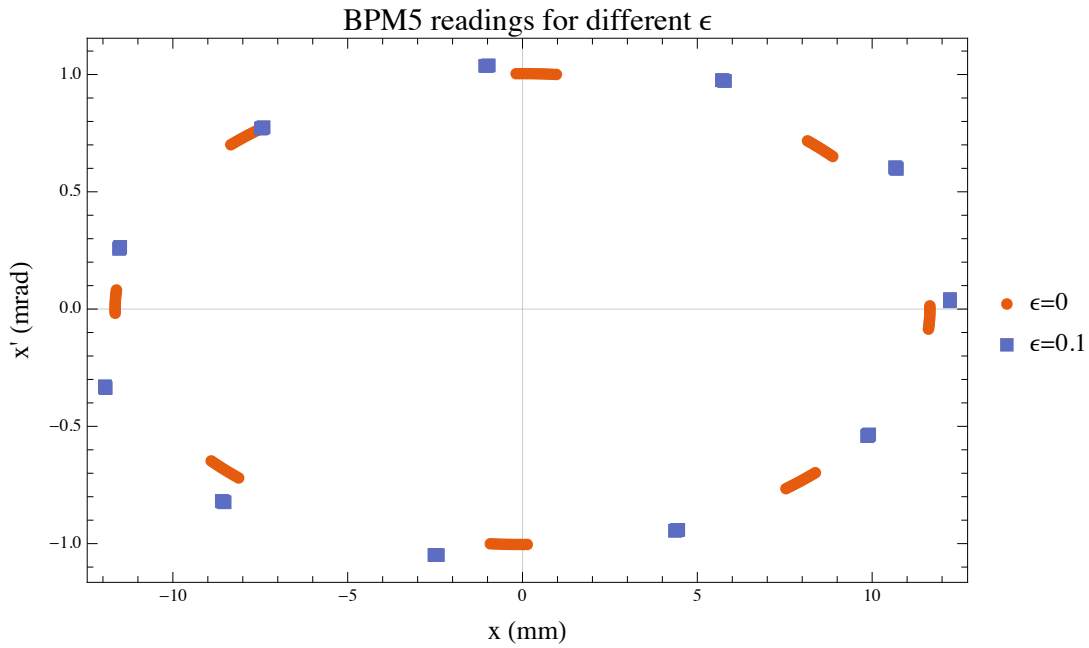
In[•]:=
```
ListPlot[{#[[1, 1]] 10^3 & /@ BPM5[x0, 100, 0], #[[1, 1]] 10^3 & /@ BPM5[x0, 100, ϵ0]},
 PlotMarkers → Automatic, Joined → True, PlotTheme → "Scientific",
 FrameLabel → {Style["Turn #", 14], Style["x (mm)", 14]},
 PlotLabel → Style["BPM5 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
ListPlot[{(#ᵀ 10^3)[[1]] & /@ BPM5[x0, 100, 0], (#ᵀ 10^3)[[1]] & /@ BPM5[x0, 100, ϵ0]},
 PlotMarkers → Automatic, PlotTheme → "Scientific",
 FrameLabel → {Style["x (mm)", 14], Style["x' (mrad)", 14]},
 PlotLabel → Style["BPM5 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=

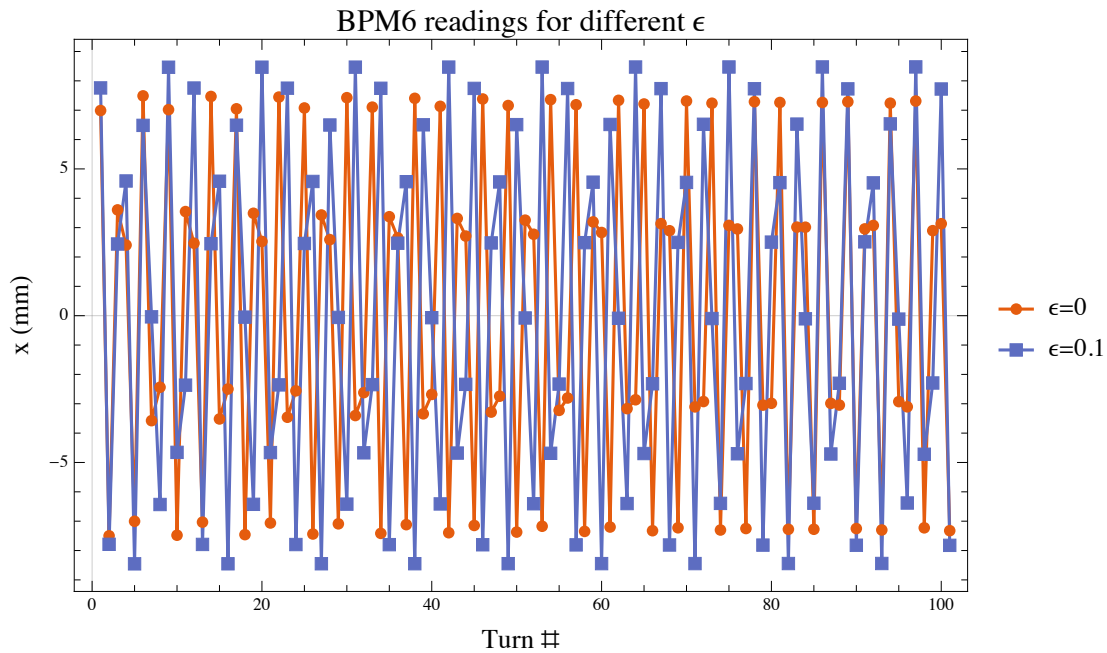


In[•]:=
```
Export[dir <> "bpm5_ideal.dat", (#ᵀ 10^3)[[1]] & /@ BPM5[x0, 100, 0]];
Export[dir <> "bpm5_err.dat", (#ᵀ 10^3)[[1]] & /@ BPM5[x0, 100, ϵ0]];
```

BPM6

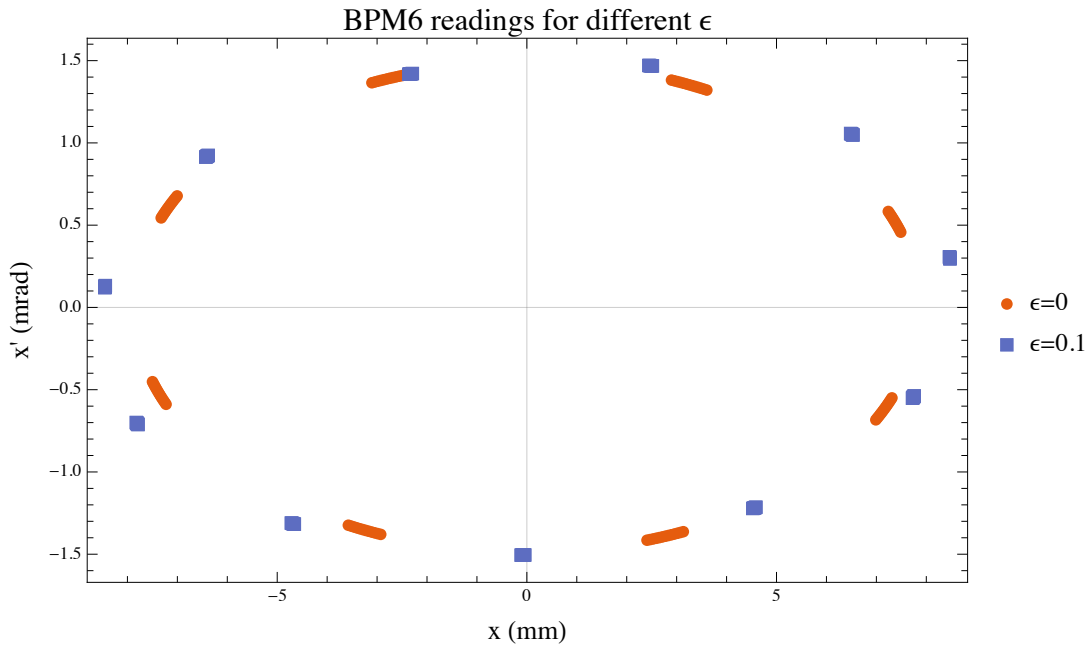
In[•]:=
```
ListPlot[{#[[1, 1]] 10^3 & /@ BPM6[x0, 100, 0], #[[1, 1]] 10^3 & /@ BPM6[x0, 100, ϵ0]},
 PlotMarkers → Automatic, Joined → True, PlotTheme → "Scientific",
 FrameLabel → {Style["Turn #", 14], Style["x (mm)", 14]},
 PlotLabel → Style["BPM6 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
ListPlot[{(#ᵀ 10³)[[1]] & /@ BPM6[x0, 100, 0], (#ᵀ 10³)[[1]] & /@ BPM6[x0, 100, ϵ0]},
 PlotMarkers → Automatic, PlotTheme → "Scientific",
 FrameLabel → {Style["x (mm)", 14], Style["x' (mrad)", 14]},
 PlotLabel → Style["BPM6 readings for different ϵ", 16],
 PlotLegends → {"ϵ=0", "ϵ=0.1"}, ImageSize → 500 ]
```

Out[•]=



In[•]:=
```
Export[dir <> "bpm6_ideal.dat", (#ᵀ 10³)[[1]] & /@ BPM6[x0, 100, 0]];
Export[dir <> "bpm6_err.dat", (#ᵀ 10³)[[1]] & /@ BPM6[x0, 100, ϵ0]];
```