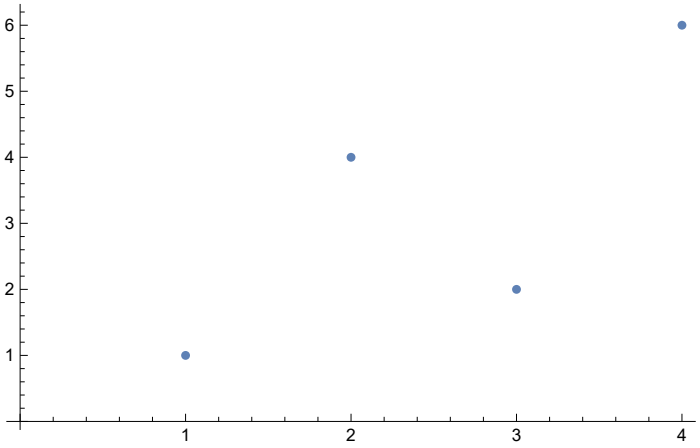
```
signal = {1, 4, 2, 6};
N = Length[signal];
ListPlot[signal]
```



FED cycle

Pad the border with 0.

$$\Delta_{s\_}[j\_] := \begin{cases} -1 * s[[j]] + s[[j+1]] & j == 1 \\ s[[j-1]] - 1 * s[[j]] & j == N; \\ s[[j-1]] - 2 * s[[j]] + s[[j+1]] & \text{True} \end{cases}$$

Function which is responsible for the varying step size depending on the current iteration i and the length of the box filter n (the total length is $2n + 1$).

$$\tau[i\_, n\_] := \frac{1}{4 * \text{Cos}\left[\pi * \frac{2*i+1}{4*n+2}\right]^2};$$

The first theorem states a similarity between the box filter and one FED cycle. Apply a box filter with $n = 1$:

```
signalBox = ListConvolve[{1/3, 1/3, 1/3},
   ArrayPad[signal, 1, "Reversed"]
  ] // N
```

```
{2., 2.33333, 4., 4.66667}
```

Now calculate one FED cycle by hand. The result is indeed the same.

```
τ[0, 1]
```

$$\frac{1}{3}$$

```
sDiv = Table[Δsignal[j], {j, 1, N}]
signal + τ[0, 1] * sDiv // N
```

```
{3, -5, 6, -4}
```

```
{2., 2.33333, 4., 4.66667}
```

Test also with a box filter of length $n = 2$. The cyclic FED seems to be identical to a reversed convolution

```
ArrayPad[signal, 2, "Reversed"]
```

```
{4, 1, 1, 4, 2, 6, 6, 2}
```

```
signalBox2 = ListConvolve[{1/5, 1/5, 1/5, 1/5, 1/5}, ArrayPad[signal, 2, "Reversed"]] // N
```

```
{2.4, 2.8, 3.8, 4.}
```

```
signal + τ[0, 2] * sDiv + τ[1, 2] * Table[Δ(signal+τ[0,2]*sDiv)[j], {j, 1, N}] // N
```

```
{2.4, 2.8, 3.8, 4.}
```

In this case two iterations are needed and the derivative of the derivative needs to be calculated (expansion is done by hand). But, the result is still identical.

```
signal + τ[0, 2] * sDiv + τ[1, 2] * sDiv + τ[0, 2] * τ[1, 2] * Table[Δ_sDiv[j], {j, 1, N}] // N
```

```
{2.4, 2.8, 3.8, 4.}
```

Matrix notation

It is also possible to write the problem in matrix-vector notation. In this case the vector is the input signal and the matrix is built in such a ways that it also encodes the second order derivative.

```
A = Table[ Piecewise[{{-1, i == 1 && j == 1 || i == N && j == N}, {-2, i == j}, {1, i == j + 1 || j == i + 1}}, 0], {i, 1, N}, {j, 1, N}];
```

```
A // MatrixForm
```

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

```
(IdentityMatrix[N] + τ[0, 1] * A).signal // N
```

```
{2., 2.33333, 4., 4.66667}
```

```
(IdentityMatrix[N] + τ[1, 2] * A).(IdentityMatrix[N] + τ[0, 2] * A).signal // N
```

```
{2.4, 2.8, 3.8, 4.}
```

Ordering doesn't matter.

```
(IdentityMatrix[N] + τ[0, 2] * A).(IdentityMatrix[N] + τ[1, 2] * A).signal // N
```

```
{2.4, 2.8, 3.8, 4.}
```

Multiple iterations:

```
FED[n_] := Module[{iter},
  iter = ConstantArray[0, n + 1];
  iter[[1]] = signal;

  Do[
   iter[[i + 2]] = (IdentityMatrix[N] + τ[i, n] * A).iter[[i + 1]]
   , {i, 0, n - 1}];

  iter
 ]
```

```
N[FED[10]]
```

```
{{1., 4., 2., 6.}, {1.75421, 2.74298, 3.50842, 4.99438}, {2.01428, 2.68424, 3.69794, 4.60354},
 {2.20757, 2.78341, 3.66675, 4.34227}, {2.39952, 2.88591, 3.59748, 4.11709},
 {2.59845, 2.978, 3.51897, 3.90458}, {2.80355, 3.06523, 3.43502, 3.6962},
 {3.00971, 3.1504, 3.34946, 3.49043}, {3.19655, 3.2279, 3.27233, 3.30322},
 {3.28677, 3.26552, 3.23337, 3.21434}, {3.04762, 3.14286, 3.38095, 3.42857}}
```

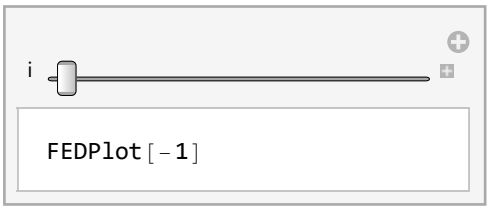
```
boxLength = 10;
signalBox10 = ListConvolve[ConstantArray[1/(2 * boxLength + 1), 2 * boxLength + 1],
    ArrayPad[signal, Floor[(2 * boxLength + 1)/2], "Reversed"]] // N
```

```
{3.04762, 3.14286, 3.38095, 3.42857}
```

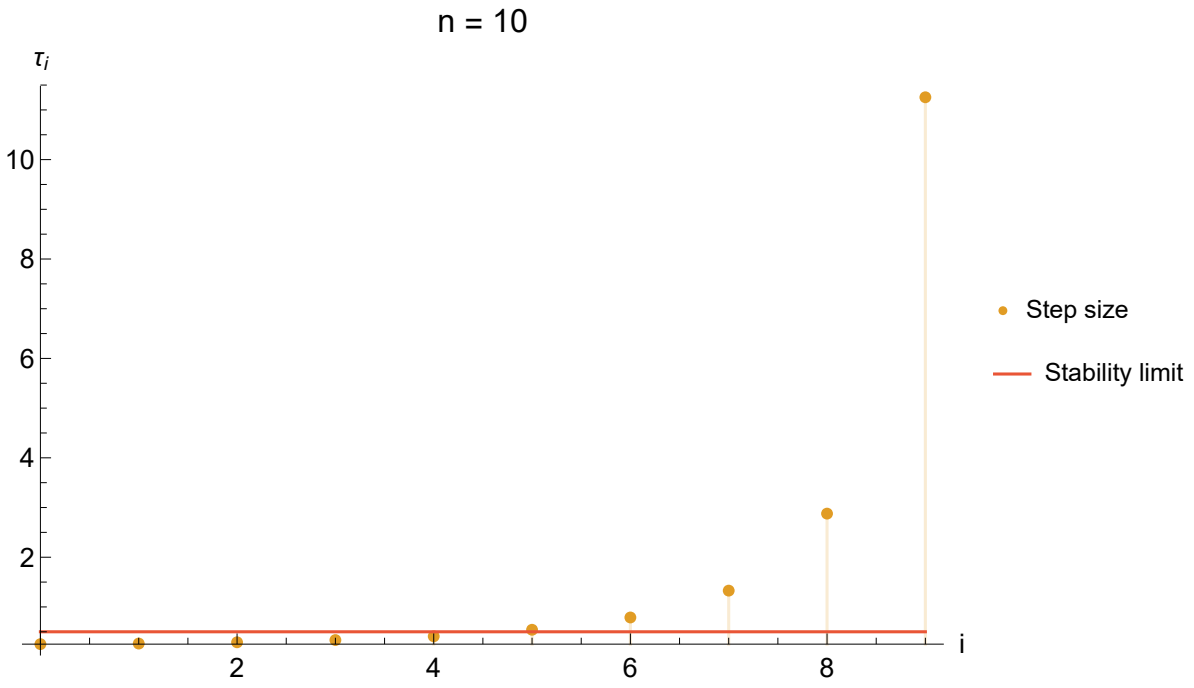
```
Table[{i, i}, {i, 1, Length[signal]}]
```

```
{{1, 1}, {2, 2}, {3, 3}, {4, 4}}
```

```
FEDPlot[i_] := Show[
   ListLinePlot[FED[10][[i + 2]] // N,
    PlotRange → {{0.9, 4.1}, {0, 6.1}},
    PlotLegends → {{None, "FED: B21·û (i = " <> IntegerString[i, 10] <> ")"}  i == -1, True},
    AxesLabel → {"uj", "*"},
    PlotStyle → {{Directive[■, Opacity[0]], i == -1}, {■, True}},
    PlotMarkers → {Graphics@{■, Disk[]}, 0.03},
    ImageSize → 500,
    BaseStyle → {FontSize → 14},
    Ticks → {Table[{j, j}, {j, 1, Length[signal]}], Automatic},
    LabelStyle → Directive[FontSize → 14]
   ],
   ListLinePlot[signalBox10,
    PlotStyle → ■,
    PlotLegends → {"Box: B̃2·n+1*û = B̃21*û"},
    PlotMarkers → {Graphics@{■, Disk[]}, 0.03},
    LabelStyle → Directive[FontSize → 14]
   ],
   ListLinePlot[signal,
    PlotStyle → Directive[Gray, Opacity[0.4]],
    PlotLegends → {"Signal û"},
    PlotMarkers → {Graphics@{Gray, Rectangle[]}, 0.03},
    LabelStyle → Directive[FontSize → 14]
   ]
  ];

Manipulate[
 FEDPlot[i]
 , {i, -1, boxLength - 1, 1}]
```



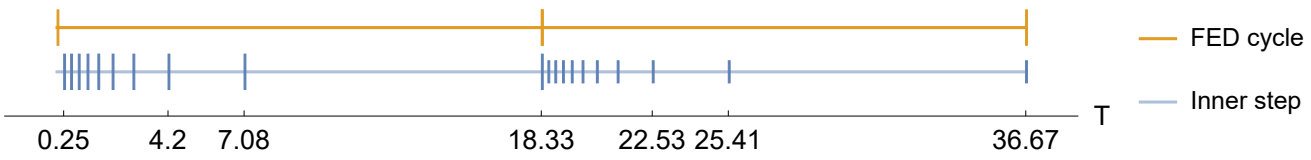

```
(*Export[FileNameJoin[{NotebookDirectory[],"frames/i=0.png"}],
  Table[FEDPlot[i],{i,-1,9}],"VideoFrames",Antialiasing→True];*)

Show[
 DiscretePlot[τ[i, boxLength], {i, 0, boxLength - 1},
  PlotLabel → "n = " <> ToString[boxLength],
  PlotRange → Full,
  ImageSize → 500,
  BaseStyle → {FontSize → 14},
  AxesLabel → {"i", "τi"},
  PlotStyle → ■,
  PlotLegends → {"Step size"}
 ],
 Plot[1/2, {x, 0, boxLength - 1}, PlotStyle → ■, PlotLegends → {"Stability limit"}]
]
```




```
cumTimes = Round[Flatten[{Table[∑_{i=0}^{n-1} τ[i, boxLength], {n, 1, 10}],
    ∑_{i=0}^{10-1} τ[i, boxLength] + Table[∑_{i=0}^{n-1} τ[i, boxLength], {n, 1, 10}]}], 0.01] // N
```

```
{0.25, 0.51, 0.8, 1.14, 1.55, 2.09, 2.87, 4.2, 7.08, 18.33,
 18.58, 18.85, 19.14, 19.47, 19.88, 20.42, 21.21, 22.53, 25.41, 36.67}
```

```
Show[
 (* Left part *)
 NumberLinePlot[{Table[Sum[τ[i, boxLength], {i, 0, n-1}], {n, 1, 10}], Interval[{0, Sum[τ[i, boxLength], {i, 0, 10-1}]}]},
   Ticks → {{cumTimes[[1]], cumTimes[[8]], cumTimes[[9]],
      cumTimes[[10]], cumTimes[[18]], cumTimes[[19]], cumTimes[[20]]}},
   ImageSize → Large,
   PlotRange → {-2, 2 + 2 * cumTimes[[10]]},
   AxesLabel → {"T"},
   BaseStyle → {FontSize → 14},
   PlotLegends → LineLegend[{None, ■}, {None, "FED cycle"}],
   LabelStyle → Directive[FontSize → 14],
   PlotStyle → {Directive[Opacity[1]], Opacity[0]}
  ] /. {Point[pos_] :> Rectangle[{pos[[1]], pos[[2]] - 0.4}, {pos[[1]] + 0.07, pos[[2]] + 0.4}]},

 (* Left yellow part *)
 NumberLinePlot[{None, Interval[{0, Sum[τ[i, boxLength], {i, 0, 10-1}]}]},
   PlotStyle → {Opacity[1], Opacity[1]}
  ] /. {Point[pos_] :> Rectangle[{pos[[1]], pos[[2]] - 0.4}, {pos[[1]] + 0.07, pos[[2]] + 0.4}]},

 (* Right blue part *)
 NumberLinePlot[{cumTimes[[10]] + Table[Sum[τ[i, boxLength], {i, 0, n-1}], {n, 1, 10}], None},
   PlotStyle → {Opacity[1], Opacity[1]}
  ] /.
  {Point[pos_] :> Rectangle[{pos[[1]], pos[[2]] - 0.25}, {pos[[1]] + 0.07, pos[[2]] + 0.25}]},

 (* Right yellow part *)
 NumberLinePlot[{None, Interval[{cumTimes[[10]], cumTimes[[10]] + Sum[τ[i, boxLength], {i, 0, 10-1}]}]},
   PlotStyle → {Opacity[1], Opacity[1]}
  ] /. {Point[pos_] :> Rectangle[{pos[[1]], pos[[2]] - 0.4}, {pos[[1]] + 0.07, pos[[2]] + 0.4}]},

 (* Blue line *)
 NumberLinePlot[{Interval[{0, cumTimes[[-1]]}]},
   PlotStyle → {Opacity[0.5]},
   PlotLegends → {"Inner step"}
  ] /. {Point[pos_] :> Rectangle[{0, 0}, {0, 0}]}

]
```



Leja Ordering

To improve numerical stability the step sizes τ_i are re-ordered. For how Leja Ordering works, see also section 2.3 from http://www.mia.uni-saarland.de/Publications/breuss-pp273.pdf

```
steps = Table[τ[i, 11], {i, 0, 10}] // N
```

```
{0.25117, 0.260795, 0.281577, 0.317122, 0.374566,
 0.468059, 0.627712, 0.926037, 1.57507, 3.43452, 13.4834}
```

```
stepsI = 1/steps
```

```
{3.98137, 3.83442, 3.55142, 3.15336, 2.66976,
 2.13648, 1.59309, 1.07987, 0.634894, 0.291161, 0.0741654}
```

```
stepsIReordered = {stepsI[[1]], stepsI[[11]], stepsI[[6]], stepsI[[8]], stepsI[[4]],
  stepsI[[10]], stepsI[[3]], stepsI[[7]], stepsI[[2]], stepsI[[9]], stepsI[[5]]}
```

```
{3.98137, 0.0741654, 2.13648, 1.07987, 3.15336,
 0.291161, 3.55142, 1.59309, 3.83442, 0.634894, 2.66976}
```

For the default ordering the values do not match.

```
Do[
 Print[∏_{k=0}^{j} Abs[stepsI[[j + 1 + 1]] - stepsI[[k + 1]]]]
 , {j, 0, Length[stepsI] - 2}]
```

```
0.146949

0.121676

0.224478

0.651323

2.40352

9.5693

35.3127

102.46

189.027

156.517
```

```
Do[
 Print[
  Max[(∏_{k=0}^{j} Abs[#1 - stepsI[[k + 1]]]) & /@ stepsI]
 ]
 , {j, 0, Length[stepsI] - 2}]
```

```
3.90721

14.6921

51.0882

157.311

408.314

842.075

1279.05

1286.34

721.289

156.517
```

But when the Leja Ordering is applied, the values do match

```
Do[
 Print[∏_{k=0}^{j} Abs[stepsIReordered[[j + 1 + 1]] - stepsIReordered[[k + 1]]]]
 , {j, 0, Length[stepsIReordered] - 2}]
```

```
3.90721

3.80475
```

```
3.08326

5.3758

3.33573

6.78523

4.02452

3.95586

9.70491

7.46948
```

```
Do[
 Print[
  Max[(∏_{k=0}^{j} Abs[#1 - stepsIReordered[[k + 1]]]) & /@ stepsIReordered]
 ]
 , {j, 0, Length[stepsIReordered] - 2}]
```

```
3.90721

3.80475

3.08326

5.3758

3.33573

6.78523

4.02452

3.95586

9.70491

7.46948
```