Accelerating Convergence

Suppose we have a fixed point method that converges linearly. Then we talked about the fact that:

\[ e_n = p - p_n \approx \frac{g'(p)}{1-g'(p)} (p_n - p_{n-1}) \]

and

\[ g'(p) \approx \frac{p_n - p_{n-1}}{p_{n-1} - p_{n-2}} \]

So, if we put these two expressions together we have that

\[ p - p_n \approx -\frac{(p_n - p_{n-1})^2}{p_n + p_{n-2} - 2p_{n-1}} \]

If this is the case then we could say that

\[ p \approx p_n - \frac{(p_n - p_{n-1})^2}{p_n + p_{n-2} - 2p_{n-1}} \]

Both Aitkens \( \Delta^2 \) method and Steffensen's method are based on using this as a better approximation.

Aitken's \( \Delta^2 \) Method

For Aitken's \( \Delta^2 \) Method, we use this fact directly and we call the new iterate \( \hat{p}_n \) to be:

\[ \hat{p}_n = p_n - \frac{(p_n - p_{n-1})^2}{p_n + p_{n-2} - 2p_{n-1}} \]
So to apply this method you start with \( p_0 \) and then find \( p_1 = g(p_0) \) and then find \( p_2 = g(p_1) \). Now you can compute \( \hat{p}_2 \), then you find \( p_3 = g(p_2) \) and then find \( \hat{p}_3 \) again using the above formula. Notice that you don't use the new iterates hat iterates to go to the next step (we will do that in Steffensen's Method). Also notice that you have to wait until \( n=2 \) to compute the new iterates, so if you compute this in your code you can add an extra little expression making sure only to compute the hatted values if \( n \geq 2 \). Let's look at an example:

```
In[26]:= g[x_] = Sqrt[10 / (2 + x)]
```

```
\[ \sqrt{10} \]
```

```
Out[26]=\[ \sqrt{10} \]
```

```
In[27]:= p_0 = 2.236
```

```
Out[27]= 2.236
```

```
In[28]:= p_1 = g[p_0]
```

```
Out[28]= 1.53646
```

```
In[29]:= p_2 = g[p_1]
```

```
Out[29]= 1.68157
```

```
In[30]:= phat_2 = p_2 - (p_2 - p_1)^2 / (p_2 + p_0 - 2 p_1)
```

```
```

```
In[31]:= p_3 = g[p_2]
```

```
Out[31]= 1.6481
```

```
In[32]:= phat_3 = p_3 - (p_3 - p_2)^2 / (p_3 + p_1 - 2 p_2)
```

```
Out[32]= 1.65437
```

Notice that I have used \( p_2 \) not \( \hat{p}_2 \) to calculate \( p_3 \).
Now let's compare the methods and the convergence. First, let's see what Mathematica will tell us about the actual value of the fixed point.

```mathematica
In[35]:=
root = FindRoot[x - g[x], {x, .4}]
```

```
Out[35]=
{x -> 1.65425}
```

```mathematica
In[36]:=
actual = x /. root
```

```
Out[36]=
1.65425
```

Now let's make some tables.

```mathematica
In[37]:=
T = Table[{n, p[n], Abs[p[n] - actual], Abs[p[n] - actual]/Abs[p[n-1] - actual], phat[n], Abs[phat[n] - actual], Abs[phat[n] - actual]/Abs[phat[n-1] - actual]}, {n, 0, 10}];
T2 = Join[{{"n", "p", "error for p", "approx Lambda", "phat", "Error for Aitken's", "Approx New Lambda"}}, T];
Grid[
T2, Frame -> All]
```

<table>
<thead>
<tr>
<th>n</th>
<th>p[n]</th>
<th>error for p[n]</th>
<th>approx Lambda</th>
<th>phat[n]</th>
<th>Error for Aitken's</th>
<th>Approx New Lambda</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.236</td>
<td>0.581751</td>
<td>0.581751</td>
<td>phat_0</td>
<td>Abs[-1.65425 + phat_0]</td>
<td>Abs[-1.65425 + phat_0] / Abs[-1.65425 + phat_0]</td>
</tr>
<tr>
<td>1</td>
<td>1.53646</td>
<td>0.117786</td>
<td>0.202469</td>
<td>phat_1</td>
<td>Abs[-1.65425 + phat_1]</td>
<td>Abs[-1.65425 + phat_1] / Abs[-1.65425 + phat_1]</td>
</tr>
<tr>
<td>2</td>
<td>1.68157</td>
<td>0.0273228</td>
<td>0.231969</td>
<td>1.65664</td>
<td>0.00239321</td>
<td>0.00239321 / Abs[-1.65425 + phat_1]</td>
</tr>
<tr>
<td>3</td>
<td>1.6481</td>
<td>0.00614994</td>
<td>0.225085</td>
<td>1.65437</td>
<td>0.000124067</td>
<td>0.051841</td>
</tr>
<tr>
<td>4</td>
<td>1.65564</td>
<td>0.00139377</td>
<td>0.226632</td>
<td>1.65426</td>
<td>6.33983 \times 10^{-6}</td>
<td>0.0511003</td>
</tr>
<tr>
<td>5</td>
<td>1.65393</td>
<td>0.000315385</td>
<td>0.226281</td>
<td>1.65425</td>
<td>3.24997 \times 10^{-7}</td>
<td>0.0512627</td>
</tr>
<tr>
<td>6</td>
<td>1.65432</td>
<td>0.0000713907</td>
<td>0.226361</td>
<td>1.65425</td>
<td>1.66482 \times 10^{-8}</td>
<td>0.0512257</td>
</tr>
<tr>
<td>7</td>
<td>1.65423</td>
<td>0.0000161588</td>
<td>0.226343</td>
<td>1.65425</td>
<td>8.52954 \times 10^{-10}</td>
<td>0.051234</td>
</tr>
<tr>
<td>8</td>
<td>1.65425</td>
<td>3.65748 \times 10^{-6}</td>
<td>0.226347</td>
<td>1.65425</td>
<td>4.36984 \times 10^{-11}</td>
<td>0.0512318</td>
</tr>
<tr>
<td>9</td>
<td>1.65425</td>
<td>8.27856 \times 10^{-7}</td>
<td>0.226346</td>
<td>1.65425</td>
<td>2.23865 \times 10^{-12}</td>
<td>0.0512297</td>
</tr>
<tr>
<td>10</td>
<td>1.65425</td>
<td>1.87382 \times 10^{-7}</td>
<td>0.226346</td>
<td>1.65425</td>
<td>1.14575 \times 10^{-13}</td>
<td>0.0511803</td>
</tr>
</tbody>
</table>
Notice that convergence is still linear, but the asymptotic rate constant has been reduced from approximately .2263 to about .0512.

**Steffensen's Method**

Steffensen's Method works very similarly, but in Steffensens Method you use the new iterate at each step when computing the next iterate.

So start with $p_0$, then to get $\hat{p}_1$, you do two sub steps which will produce $p_{0,1}$ and $p_{0,2}$. First you use the iteration function twice to get these preliminary values.

$p_{1,0} = g(p_0)$ and $p_{2,0} = g(p_{1,0})$. Then you use these two values to compute $\hat{p}_1$ using the formula:

$$\hat{P}_1 = p_{2,0} - \frac{(p_{2,0}-p_{1,0})^2}{p_{2,0}+\hat{p}_0-2p_{1,0}}$$

The general formula is then $p_{1,n} = g(\hat{p}_n)$ and $p_{2,n} = g(p_{1,n})$ and:

$$\hat{P}_{n+1} = p_{2,n} - \frac{(p_{2,n}-p_{1,n})^2}{p_{2,n}+\hat{p}_n-2p_{1,n}}$$

```plaintext
In[40]:= phatS0 = p0;
Do[{p1n = g[phatSn], p2n = g[p1n],
    phatSn+1 = p2n - (p2n - p1n)^2 / (p2n - 2 p1n + phatSn)}, {n, 0, 10}]
```
The last column of the above table is $|e_n|/|e_{n-1}|^2$ and indicates that at least in this case Steffensen's method converges quadratically. This is the case in general, if you begin with an iteration scheme that converges linearly and use Steffensen's Method it will converge quadratically.