Continuous Newton’s Method for Power Flow Analysis

Dr. Federico Milano

E-mail: Federico.Milano@uclm.es

Department of Electrical Engineering

University of Castilla - La Mancha

Spain
Biography

Federico Milano received from the University of Genoa, Italy, the Electrical Engineering degree and the Ph.D. degree in 1999 and 2003, respectively.

In 2001 and 2002, he worked at the University of Waterloo, Canada, as a Visiting Scholar.

He is currently a tenure associate professor at the University of Castilla-La Mancha, Ciudad Real, Spain.

His research interests include voltage stability, electricity markets and computer-based power system modelling.
Contents

- Power Flow Problem
- Existing Power Flow Solution Methods
- Continuous Newton’s Method
- Conclusions
Power Flow Problem
A classical problem of circuit theory is to find all branch currents and all node voltages of an assigned circuit.

Typical input data are generator voltages as well as the impedances of all branches.

If all impedances are constant, the resulting set of equations that describe the circuit is linear.
Example 1 (I)

Classical circuit problem.

![Circuit Diagram]
Example 1 (II)

Using the branch current method, one has:

\[ 0 = \frac{\bar{v}_1 - \bar{v}_2}{jx_{12}} + \frac{\bar{v}_1 - \bar{v}_3}{jx_{13}} - \bar{i}_1 \]  (1)

\[ 0 = \frac{\bar{v}_2 - \bar{v}_1}{jx_{12}} + \frac{\bar{v}_2 - \bar{v}_3}{jx_{23}} - \bar{i}_2 \]

\[ 0 = \frac{\bar{v}_3 - \bar{v}_1}{jx_{13}} + \frac{\bar{v}_3 - \bar{v}_2}{jx_{23}} - \bar{i}_3 \]
Example 1 (III)

In vector form:

\[
\begin{bmatrix}
\vec{i}_1 \\
\vec{i}_2 \\
0
\end{bmatrix}
= \left( \bar{Y} + I_3 \begin{bmatrix}
0 & 0 \\
0 & 1/\bar{z}_3
\end{bmatrix} \right)
\begin{bmatrix}
\vec{v}_1 \\
\vec{v}_2 \\
\vec{v}_3
\end{bmatrix}
= \bar{Y}_{\text{tot}} \vec{v}
\] (2)
Example 1 (IV)

where $I_3$ is a $3 \times 3$ identity matrix and $\bar{Y}$ is the so-called admittance matrix:

$$
\bar{Y} = \begin{bmatrix}
\frac{1}{j x_{12}} + \frac{1}{j x_{13}} & -\frac{1}{j x_{12}} & -\frac{1}{j x_{13}} \\
-\frac{1}{j x_{12}} & \frac{1}{j x_{12}} + \frac{1}{j x_{23}} & -\frac{1}{j x_{23}} \\
-\frac{1}{j x_{13}} & -\frac{1}{j x_{23}} & \frac{1}{j x_{13}} + \frac{1}{j x_{23}}
\end{bmatrix}
$$

(3)
Background (II)

- The power flow problem is conceptually the same problem as solving a steady-state ac circuit.

- The only, though substantial, difference is the set of input data.

- Loads are expressed in terms of consumed active and reactive powers \((PQ\ load)\) and generators are defined in terms of constant voltage magnitude and active power injection \((PV\ generator)\).

- Hence, the power flow problem is **nonlinear**.
Example 2 (I)

Classical power flow problem.
Example 2 (II)

The power flow problem is formulated in order to determine unknown voltage magnitudes and angles.

\[
0 = \frac{v_2 v_1}{x_{12}} \sin(\theta_2 - \theta_1) - p_2
\]

\[
0 = \frac{v_3 v_1}{x_{13}} \sin(\theta_3 - \theta_1) + \frac{v_3 v_2}{x_{23}} \sin(\theta_3 - \theta_2) - p_3
\]

\[
0 = \frac{v_3^2}{x_{13}} + \frac{v_3^2}{x_{23}} - \frac{v_3 v_1}{x_{13}} \cos(\theta_3 - \theta_1) - \frac{v_3 v_2}{x_{23}} \cos(\theta_3 - \theta_2) - q_3
\]

where the unknowns are \( v_3, \theta_3 \) and \( \theta_2 \).
Rationales Behind Power Flow Problem Formulation

- Loads at high voltage level are modelled as constant PQ due to under-load tap changers.
- Generators are modelled as PV due to turbine and voltage regulators.
- Transmission lines and transformers are generally modelled as lumped $\pi$-circuits with constant parameters.
- Observe that one bus has to be the phase angle reference. Typically, one generator is used as slack bus.
- The distributed slack bus model is more physical.
The vector of currents injected at each node is:

\[ \bar{i} = \bar{Y} \bar{v} \]  \hspace{1cm} (5)

which leads to write the power flow problem as the complex power injections at buses:

\[ \bar{s} = \bar{V} \bar{i}^* = \bar{V} \bar{Y}^* \bar{v}^* \]  \hspace{1cm} (6)

where \( \bar{V} = \text{diag}(\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{n_b}) \) and \( n_b \) is the number of network buses.
General Formulation of the Power Flow Problem (II)

Rewriting (6):

$$0 = \bar{s} - \bar{V} \bar{Y}^* \bar{v}^*$$

(7)

Or, more in general:

$$0 = g(y)$$

(8)

where $y$ and data per bus type are:

<table>
<thead>
<tr>
<th>Bus type</th>
<th>Variables</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slack generator</td>
<td>$p, q$</td>
<td>$v, \theta$</td>
</tr>
<tr>
<td>PV generator</td>
<td>$q, \theta$</td>
<td>$p, v$</td>
</tr>
<tr>
<td>PQ load</td>
<td>$v, \theta$</td>
<td>$p, q$</td>
</tr>
</tbody>
</table>
Relevant Issues of the Power Flow Problem

The origins of the formulation of the power flow problem and the solution based on the Newton’s method date back to the late sixties.

Since then, a huge variety of studies have been presented about the solution of the power flow problem, addressing:

- Starting initial guess
- Computational efficiency
- Ill-conditioned cases
- Robustness
- Multiple solutions
- Unsolvable cases
Taxonomy of the Power Flow Problem

It is relevant to classify the power flow problems into the following categories:

- Well-conditioned case
- Ill-conditioned case
- Bifurcation point
- Non-physical solution
- Unsolvable case
Well-conditioned case

The power flow solution exists and is reachable using a flat initial guess (e.g., all load voltage magnitudes equal to 1 and all bus voltage angles equal to 0) and a standard Newton’s method.

This case is the most common situation.
Ill-conditioned case

- The solution of the power flow problem does exist, but standard solvers fail to get this solution starting from a flat initial guess.

- This situation is due to the fact that the region of attraction of the power flow solution is narrow or does not contain the initial guess.

- In this case, the failure of standard power flow procedure is due to the instability of the numerical method, not of the power flow equations.

- Robust power flow methods have proved to be efficacious for solving ill-conditioned cases.
Bifurcation Point

The solution of the power flow exists but it is either a saddle-node bifurcation or a limit-induced bifurcation.

- Saddle-node bifurcations are associated with the maximum loading condition of a system. The solution cannot be obtained using standard or robust power flow methods, since the power flow Jacobian matrix is singular at the solution point.

- Limited-induced bifurcations are associated with a physical limit of the system, such as the shortage of generator reactive power. The solution point is typically a well-conditioned case and does not show convergence issues.
Non-physical Solution

- Solutions that cannot be accepted since some variable is out of its technical limits.

- Typically these solutions are characterized by very low voltage levels.

- These solutions are also known as: extraneous, false, lower, or unstable.
Unsolvable Case

- The solution of the power flow problem does not exist.

- Typically, the issue is that the loading level of the network is too high.

- As in the case of the bifurcation points, a continuation method or an optimal power flow problem allows defining the maximum loading level that the system can supply.
Existing Power Flow Solution Methods
Solution Methods (I)

Methods that do not require the computation of the Jacobian matrix of $g$:

- Jacobi’s method.
- Gauss-Seidel’s method.

Methods that require the computation of the Jacobian matrix of $g$:

- Newton’s (or Newton-Raphson’s) method.
- Robust Newton’s methods.
Solution Methods (II)

Methods that simplify the Jacobian matrix of $g$:

- Inexact and dishonest Newton’s methods.
- Fast decoupled power flow.

Methods that simplify $g$:

- DC power flow model.
Newton-Raphson’s Method (I)

The \(i\)-th iteration of the classical Newton’s method for (8) is as follows:

\[
\Delta y^{(i)} = -[g_y^{(i)}]^{-1} g^{(i)}
\]

\[
y^{(i+1)} = y^{(i)} + \Delta y^{(i)}
\]

where \(g^{(i)} = g(y^{(i)})\), \(g_y^{(i)} = g_y(y^{(i)})\), and \(g_y = \nabla_T y g\) is the Jacobian matrix of the power flow equations.
Newton-Raphson’s Method (II)

The geometrical interpretation of the Newton’s method is well-known. For the actual value $y^{(i)}$, one computes the tangent of $g^{(i)}$ as:

$$
\tau(y) = g^{(i)} + g_y^{(i)}(y - y^{(i)})
$$

Imposing $\tau(y) = 0$ yields the value $y^{(i+1)}$ defined in (9).
Geometrical Interpretation of the Newton-Raphson’s Method

(a) 

(b) 

(c) 

(d)
Robust Newton’s Method (I)

- There are idiosyncratic cases for which the Newton’s technique fails to converge.

- A variety of robust variations of the basic Newton’s method have been proposed in the literature for solving ill-conditioned cases.

- The majority of these techniques mainly consist in modifying the first equation of (9) as follows:

\[ \Delta y^{(i)} = -\alpha [g_y^{(i)}]^{-1} g^{(i)} \]  

(11)

where \( \alpha \) is a factor that improves the convergence properties of the iterative process.
Robust Newton’s Method (II)

- If $\alpha$ is the result of an optimization process, $\alpha$ is called *optimal multiplier* (e.g., Iwamoto’s method).

- It is important not to confuse ill-conditioned cases with those that are unsolvable since the solution does not exist.

- Robust solvers are useful in case of ill-conditioned systems but do not generally work well for unsolvable cases.

- Unsolvable cases are better tackled using the continuation power flow technique.
Geometrical Interpretation of the Robust Newton’s Method
Main Issue of the Newton’s Method

- One of the most relevant drawbacks of the Newton’s method is the need of factorizing the full Jacobian matrix at each iteration.

- From the computational point of view, the factorization of a matrix is an order $N^3$ operation, i.e., the computational weight increases with the cube of the size $N$ of the matrix.

- The computational effort can be reduced to $N^{1.5}$ if using sparse matrices techniques, which allows saving a considerable time for large systems (e.g., thousands of buses).

- However, the Jacobian matrix factorization remains the most critical issue of the Newton’s method (about 85% of the total CPU time for networks with thousands of buses).
Inexact and Dishonest Newton’s Method

- Inexact methods aim to approximate the factorization of the Jacobian matrix.

- A family of inexact methods are based on the Generalized Minimal Residual (GMRES) method.

- The GMRES is a particular case of Krylov’s subspace. The main issue is to properly pre-conditioning the Jacobian matrix.

- Dishonest methods compute the Jacobian matrix factorization only the first one or two iterations and then use the previous factorization for the remaining iterations.
Geometrical Interpretation of the Dishonest Newton’s Method

(a) $\angle \tilde{g}_y$

(b) $\angle \tilde{g}_y$
Fast Decoupled Power Flow (FDPF)

- The FDPF is a particular case of dishonest Newton’s method.
- The Jacobian matrix is approximated so that it becomes block diagonal and all non-zero elements are constant. Hence only one factorization is needed.
- The FDPF requires much more iterations than the NR method but proved to be more robust.
Comparison of Power Flow Solution Methods (I)

<table>
<thead>
<tr>
<th>Bus #</th>
<th>Newton</th>
<th>Jacobi</th>
<th>Gauss-Seidel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter. #</td>
<td>time [s]</td>
<td>Iter. #</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>0.0050</td>
<td>76</td>
</tr>
<tr>
<td>118</td>
<td>5</td>
<td>0.0287</td>
<td>580</td>
</tr>
<tr>
<td>1228</td>
<td>5</td>
<td>0.210</td>
<td>454</td>
</tr>
<tr>
<td>11856</td>
<td>4</td>
<td>3.15</td>
<td>340</td>
</tr>
</tbody>
</table>
## Comparison of Power Flow Solution Methods (II)

<table>
<thead>
<tr>
<th>Bus #</th>
<th>GMRES</th>
<th></th>
<th>Dishonest</th>
<th></th>
<th>FDPF</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter. #</td>
<td>time [s]</td>
<td>Iter. #</td>
<td>time [s]</td>
<td>Iter. #</td>
<td>time [s]</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>0.4339</td>
<td>7</td>
<td>0.0040</td>
<td>6</td>
<td>0.0053</td>
</tr>
<tr>
<td>118</td>
<td>7</td>
<td>53.53</td>
<td>15</td>
<td>0.0183</td>
<td>6</td>
<td>0.0117</td>
</tr>
<tr>
<td>1228</td>
<td>n.a.</td>
<td>n.a.</td>
<td>26</td>
<td>0.207</td>
<td>12</td>
<td>0.160</td>
</tr>
<tr>
<td>11856</td>
<td>n.a.</td>
<td>n.a.</td>
<td>10</td>
<td>3.820</td>
<td>5</td>
<td>5.174</td>
</tr>
</tbody>
</table>
Region of Attraction (I)

- A key issue of any iterative technique is the initial guess.

- The only way to know if a given initial guess is adequate for obtaining a solution $y_0$ of the power flow problem is to determine the region of attraction of $y_0$. At this regard, the initial guess can be of two types:
  - The initial guess is inside the region of attraction of the solution $y_0$ and the numerical method converges.
  - The initial guess is outside the region of attraction of the solution $y_0$. By definition, a numerical method diverges or converges to a non-physical solution if one starts with such initial guess.
Example 3 (I)

Consider the following simple 2-bus system.

\[ \bar{v}_1 = 1.0 + j0 \text{ pu} \]

\[ \bar{s}_2 = 0.9 + j0.6 \text{ pu} \]

\[ \bar{z} = 0.01 + j0.1 \text{ pu} \]
Example 3 (II)

Region of attraction of the Newton’s method for a 2-bus system.
Region of Attraction (II)

- Different methods have different regions of attraction.
- Unfortunately defining the region of attraction is extremely costly.
- In practice it is virtually impossible to define the region of attraction for a real-world system.
- Robust methods can be thought as a way of enlarging the region of attraction.
Continuous Newton’s Method
Ordinary Differential Equations (ODE)

Let us consider a set of autonomous ODE:

\[ \dot{y} = \tilde{f}(y) \]  \hspace{1cm} (12)

The simplest method of numerical integrating (12) is the explicit Euler’s method, as follows:

\[ \Delta y^{(i)} = \Delta t \tilde{f}(y^{(i)}) \]  \hspace{1cm} (13)

\[ y^{(i+1)} = y^{(i)} + \Delta y^{(i)} \]

where \( \Delta t \) is a given step length.
Newton-Raphson’s Method (reprise)

Let’s recall the $i$-th iteration of the classical Newton’s method:

$$
\Delta y^{(i)} = -[g_y^{(i)}]^{-1} g^{(i)}
$$

$$
y^{(i+1)} = y^{(i)} + \Delta y^{(i)}
$$

where $g^{(i)} = g(y^{(i)})$, $g_y^{(i)} = g_y(y^{(i)})$, and $g_y = \nabla^T_y g$ is the Jacobian matrix of the power flow equations.
Analogy between Euler’s and Newton’s Methods

The analogy between (14) and (13) is straightforward if one defines:

\[ \tilde{f}(y) = -[g_y]^{-1}g(y) \]  

(15)

Equations (14) can thus be viewed as the \( i^{\text{th}} \) step of the Euler’s forward method where \( \Delta t = 1 \).

Furthermore, robust Newton’s methods are nothing but the \( i^{\text{th}} \) step of the Euler’s integration method where \( \Delta t = \alpha \).

In other words, the computation of the optimal multiplier \( \alpha \) corresponds to a variable step forward Euler’s method.
Continuous Newton’s Method

Equations (12) and (15) leads to:

\[ \dot{y} = -[g_y]^{-1}g(y) \]  \hspace{1cm} (16)

which is known as continuous Newton’s method.

The equilibrium point \( y_0 \) of (16) is

\[ 0 = \tilde{f}(y_0) = -[g_y|0]^{-1}g(y_0) \]  \hspace{1cm} (17)

Thus, assuming that \( g_y \) is not singular, \( y_0 \) is also the solution of the power flow problem.
Stability of the Continuous Newton’s Method (I)

Differentiating (15) with respect to $y$ leads to:

$$\tilde{f}_y = \nabla^T_y \tilde{f}(y)$$  \hspace{1cm} (18)

$$= -[g_y]^{-1}g_y - (\nabla^T_y ([g_y]^{-1}))g(y)$$

$$= -I_{n_y} - (\nabla^T_y ([g_y]^{-1}))g(y)$$

where $I_{n_y}$ is the identity matrix of order $n_y$. Since the equilibrium point $y_0$ is a solution for $g(y_0) = 0$, one has:

$$\tilde{f}_y|_0 = -I_{n_y}$$  \hspace{1cm} (19)
Stability of the Continuous Newton’s Method (II)

- Equation (19) implies that all eigenvalues of \( \tilde{f}_y \) at the solution point are equal to \(-1\).

- (19) means that the solution of (16), if exists, is asymptotically stable.

- The reachability of this solution depends on the starting point
  \( y(t_0) = y^{(0)} \), which has to be within the region of attraction or stability region of the equilibrium point \( y_0 \).

- The continuous Newton’s method is expected to show better ability to converge than other methods previously discussed if the initial guess is within the region of attraction.
Example 4

The previous result is straightforward for a scalar $g(y)$, i.e. for $y \in \mathbb{R}$ and $g \in \mathbb{R}$, as follows:

\[
\dot{y} = \tilde{f}(y) = -\frac{g(y)}{g_y(y)}
\]  

\[
\Rightarrow \quad \tilde{f}_y(y) = -\frac{g_y(y)}{g_y(y)} + \frac{g_{yy}(y)}{g_y(y)^2} g(y)
\]

\[
= -1 + \frac{g_{yy}(y)}{g_y(y)^2} g(y)
\]

thus $\tilde{f}_y(y_0) = -1$ if $g(y_0) = 0$ and $g_y(y_0) \neq 0$. 
It is well-known that the forward Euler’s method, even with variable time step, can be numerically unstable.

Given the analogy between the power flow equations and an ODE system, any well-assessed numerical method can be used for integrating (16).

It is thus intriguing to use some efficient integration method for solving (16).
Runge-Kutta Formulas

Since computing the Jacobian matrix of (16) is complex, explicit ODE integration methods should be preferred.

Runge-Kutta formulas are good candidates.

For example the classic RK4 formula is as follows:

\[
\begin{align*}
    k_1 &= f(x^{(i)}) \\
    k_2 &= f(x^{(i)} + 0.5\Delta tk_1) \\
    k_3 &= f(x^{(i)} + 0.5\Delta tk_2) \\
    k_4 &= f(x^{(i)} + \Delta tk_3) \\
    x^{(i+1)} &= x^{(i)} + \Delta t(k_1 + 2k_2 + 2k_3 + k_4)/6
\end{align*}
\]
Example 5 (I)

Let consider the 1254-bus model of the UCTE system.
Example 5 (II)

Comparison of methods for solving the power flow of the UCTE system.

<table>
<thead>
<tr>
<th>Method</th>
<th>$#\text{ Iter. }$ $\epsilon = 10^{-3}$</th>
<th>$#\text{ Iter. }$ $\epsilon = 10^{-4}$</th>
<th>$#\text{ Iter. }$ $\epsilon = 10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard NR</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fast Decoupled PF</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Iwamoto’s method</td>
<td>99</td>
<td>320</td>
<td>1021</td>
</tr>
<tr>
<td>Simple robust method</td>
<td>31</td>
<td>39</td>
<td>47</td>
</tr>
<tr>
<td>Runge-Kutta method</td>
<td>10</td>
<td>13</td>
<td>16</td>
</tr>
</tbody>
</table>

where $\epsilon$ is the required convergence error tolerance.
Example 5 (III)

Comparison of convergence errors obtained with different robust power flow solution methods for the UCTE system.
Example 5 (IV)

Comparison of the computational burden of different solution techniques for the UCTE system.

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iwamoto’s method</td>
<td>106.5</td>
</tr>
<tr>
<td>Simple robust method</td>
<td>3.5</td>
</tr>
<tr>
<td>Runge-Kutta method</td>
<td>3.4</td>
</tr>
</tbody>
</table>
Conclusions
Contributions

This talk has presented a continuous version of the Newton’s method for solving the power flow problem.

The main contributions discussed in the talk are:

- A general framework for applying efficient numerical integration techniques for solving ill-conditioned power flow cases.

- Proposal of a formal taxonomy of the existing numerical methods for solving the power flow problem.
Future Work

Future work will concentrate on further developing the analogy between the power flow problem, ODE systems and homotopy methods.

The stability and region of attraction of the continuous Newton’s method are promising fields of research.
References

Further insights and references to all topics covered in the talk can be found in:


Available at:

http://www.uclm.es/area/gsee/Web/Federico/publications.htm
Thanks for your attention!
Questions?