Cross-points in the Neumann-Neumann method

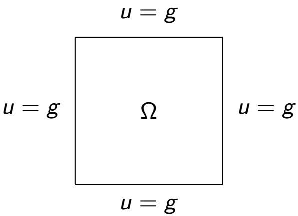
Bastien Chaudet-Dumas (University of Geneva)
Martin J. Gander (University of Geneva)

CANUM, Evian-Les-Bains

June, 14th 2022

Poisson problem on a square

We consider the simplest form of elliptic problem on the square $\Omega := (-1,1) \times (-1,1)$, with Dirichlet boundary conditions on $\partial\Omega$.



Given a source term $f \in L^2(\Omega)$, and $g \in H^{\frac{1}{2}}(\partial\Omega)$, find u solution to :

$$\begin{cases} -\Delta u = f \text{ in } \Omega\,, \\ \quad u = g \text{ on } \partial\Omega\,. \end{cases} \tag{P}$$

Domain decomposition

Principle : The domain is decomposed into a collection of smaller subdomains $\{\Omega_i\}_i$, which enables us to solve the PDE separately in each subdomain.

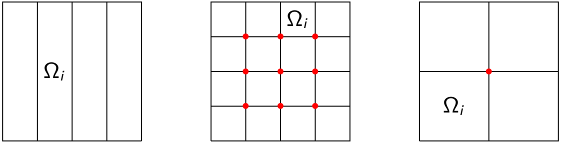


Figure – Examples of non-overlapping domain decompositions for Ω.

$\longrightarrow$ This leads to a collection of coupled problems on smaller subdomains.
$\longrightarrow$ We use an iterative process based on *transmission conditions* to decouple these subproblems.
$\longrightarrow$ Once decoupled, subproblems can be solved in parallel (*divide and conquer*).

Remark. Marked points are referred to as "cross-points".

Domain decomposition

Principle : The domain is decomposed into a collection of smaller subdomains $\{\Omega_i\}_i$, which enables us to solve the PDE separately in each subdomain.

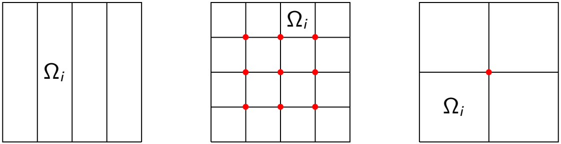


Figure – Examples of non-overlapping domain decompositions for Ω.

$\longrightarrow$ This leads to a collection of coupled problems on smaller subdomains.
$\longrightarrow$ We use an iterative process based on *transmission conditions* to decouple these subproblems.
$\longrightarrow$ Once decoupled, subproblems can be solved in parallel (*divide and conquer*).

Remark. Marked points are referred to as "cross-points".

Domain decomposition

Principle : The domain is decomposed into a collection of smaller subdomains $\{\Omega_i\}_i$, which enables us to solve the PDE separately in each subdomain.

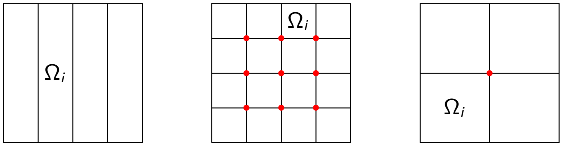


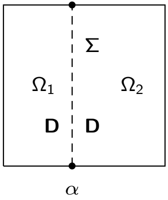
Figure – Examples of non-overlapping domain decompositions for Ω.

$\longrightarrow$ This leads to a collection of coupled problems on smaller subdomains.
$\longrightarrow$ We use an iterative process based on *transmission conditions* to decouple these subproblems.
$\longrightarrow$ Once decoupled, subproblems can be solved in parallel (*divide and conquer*).

Remark. Marked points are referred to as "cross-points".

The Neumann-Neumann method

Two subdomain version : The domain Ω is split into two subdomains by means of a vertical interface Σ placed at $x = \alpha$. Given an initial guess $\lambda^0 \in H^{\frac{1}{2}}(\Sigma)$ and a relaxation parameter $\theta \in \mathbb{R}$, each iteration $k \geq 1$ of the NNM can be decomposed in three steps :

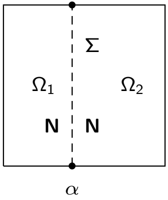


- **(Dirichlet step)** Solve in each subdomain Ω_i,

$$\begin{cases} -\Delta u_i^k = f \ \text{ in } \Omega_i \,, \\ u_i^k = g \ \text{ on } \partial\Omega_i \cap \partial\Omega \,, \\ u_i^k = \lambda^{k-1} \ \text{ on } \Sigma \,. \end{cases}$$

The Neumann-Neumann method

Two subdomain version : The domain Ω is split into two subdomains by means of a vertical interface Σ placed at $x = \alpha$. Given an initial guess $\lambda^0 \in H^{\frac{1}{2}}(\Sigma)$ and a relaxation parameter $\theta \in \mathbb{R}$, each iteration $k \geq 1$ of the NNM can be decomposed in three steps :



- **(Neumann step)** Compute the correction ψ in each subdomain Ω_i,

$$\begin{cases} -\Delta\psi_i^k = 0 \ \text{ in } \Omega_i \ , \\ \psi_i^k = 0 \ \text{ on } \partial\Omega_i \cap \partial\Omega \ , \\ \partial_{n_i}\psi_i^k = \partial_{n_i}u_i^k + \partial_{n_j}u_j^k \ \text{ on } \Sigma \ . \end{cases}$$

The Neumann-Neumann method

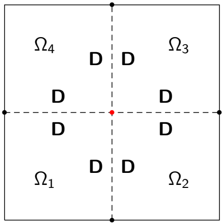
Two subdomain version : The domain Ω is split into two subdomains by means of a vertical interface Σ placed at $x = \alpha$. Given an initial guess $\lambda^0 \in H^{\frac{1}{2}}(\Sigma)$ and a relaxation parameter $\theta \in \mathbb{R}$, each iteration $k \geq 1$ of the NNM can be decomposed in three steps :

- **(Update λ)** Update the trace on Σ using the relaxation parameter,

$$\lambda^k = \lambda^{k-1} - \theta(\psi_1^k + \psi_2^k)\,.$$

The Neumann-Neumann method

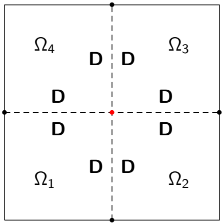
Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

The Neumann-Neumann method

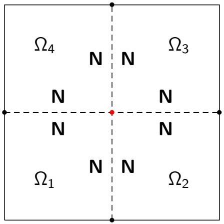
Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

The Neumann-Neumann method

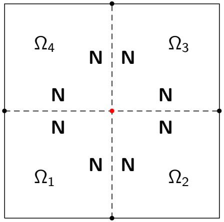
Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

The Neumann-Neumann method

Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

Even/odd splitting

Definition 1. *Any function $h \in L^p(\Omega)$, with $p \in [1, \infty]$, can be uniquely decomposed as $h = h_e + h_o$, where h_e and h_o are called the even symmetric and odd symmetric parts of h, respectively, and satisfy for almost all $(x, y) \in \Omega$*

$$h_e(-x, -y) = h_e(x, y) \,,$$
$$h_o(-x, -y) = -h_o(x, y) \,.$$

Even/odd splitting

Definition 1. *Any function $h \in L^p(\Omega)$, with $p \in [1, \infty]$, can be uniquely decomposed as $h = h_e + h_o$, where h_e and h_o are called the even symmetric and odd symmetric parts of h, respectively, and satisfy for almost all $(x, y) \in \Omega$*

$$h_e(-x, -y) = h_e(x, y) \, ,$$
$$h_o(-x, -y) = -h_o(x, y) \, .$$

Example : $h(x, y) = (2y^3 - y^2)e^{-x^2}$.

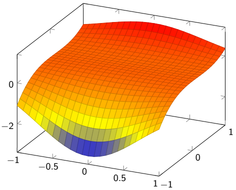


Figure – Graph of h on Ω.

Even/odd splitting

Definition 1. *Any function $h \in L^p(\Omega)$, with $p \in [1, \infty]$, can be uniquely decomposed as $h = h_e + h_o$, where h_e and h_o are called the even symmetric and odd symmetric parts of h, respectively, and satisfy for almost all $(x, y) \in \Omega$*

$$h_e(-x, -y) = h_e(x, y)\,,$$
$$h_o(-x, -y) = -h_o(x, y)\,.$$

Example : $h(x, y) = (2y^3 - y^2)e^{-x^2}$.

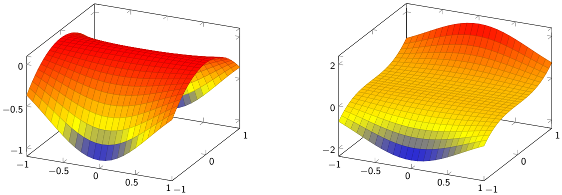


Figure – Graphs of h_e (left) and h_o (right) on Ω.

Convergence analysis - Outline (even symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_e).

Find u_e solution to :

$$\begin{cases} -\Delta u_e = f_e \text{ in } \Omega \,, \\ \quad u_e = g_e \text{ on } \partial\Omega \,. \end{cases} \tag{P_e}$$

$\longrightarrow$ At each iteration of the NNM, all local problems are well-posed.

$\longrightarrow$ Local corrections in subdomains Ω_2, Ω_3 and Ω_4 can be deduced from the local correction in Ω_1 using symmetry arguments.

$\longrightarrow$ The algorithm converges *geometrically* to the real solution u_e.

Convergence analysis - Outline (even symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_e).

Numerical illustration with $f_e = 1$, $g_e = 0$ and $\lambda_e^0 = 0$, for $\theta = 0.15$.

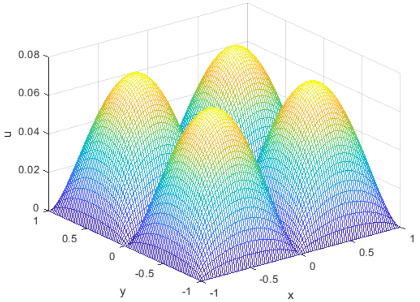


Figure – Solution u_e^1 at iteration 1, $\theta = 0.15$.

Convergence analysis - Outline (even symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_e).

Numerical illustration with $f_e = 1$, $g_e = 0$ and $\lambda_e^0 = 0$, for $\theta = 0.15$.

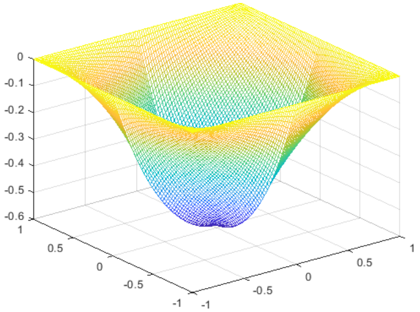


Figure – Correction ψ_e^1 at iteration 1, $\theta = 0.15$.

Convergence analysis - Outline (even symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_e).

Numerical illustration with $f_e = 1$, $g_e = 0$ and $\lambda_e^0 = 0$, for $\theta = 0.15$.

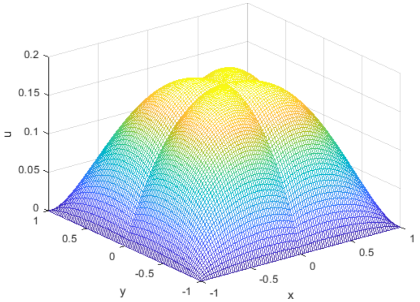


Figure – Solution u_e^2 at iteration 2, $\theta = 0.15$.

Convergence analysis - Outline (even symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_e).

Numerical illustration with $f_e = 1$, $g_e = 0$ and $\lambda_e^0 = 0$, for $\theta = 0.15$.

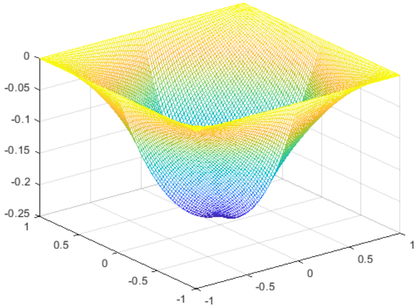


Figure – Correction ψ_e^2 at iteration 2, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_o).

Find u_o solution to :

$$\begin{cases} -\Delta u_o = f_o \text{ in } \Omega\,, \\ \quad\; u_o = g_o \text{ on } \partial\Omega\,. \end{cases} \qquad (P_o)$$

$\longrightarrow$ At some point, the Dirichlet conditions enforced in Ω_1 and Ω_3 become discontinuous.

$\longrightarrow$ Well-posedness is no longer guaranteed. Singular solutions are generated and they propagate through the following iterations, which prevents the NNM from converging.

Convergence analysis - Outline (odd symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.45$.

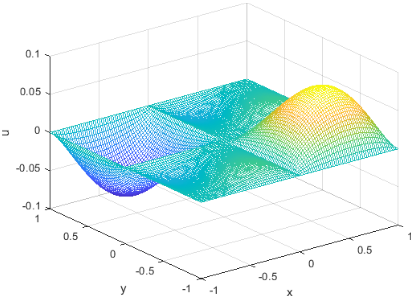


Figure – Solution u_o^1 at iteration 1, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

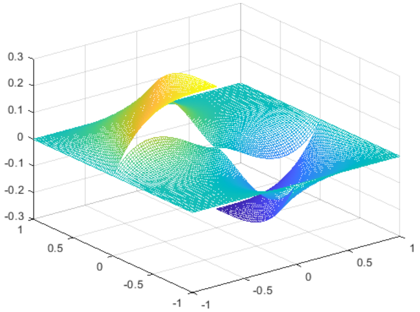


Figure – Correction ψ_o^1 at iteration 1, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

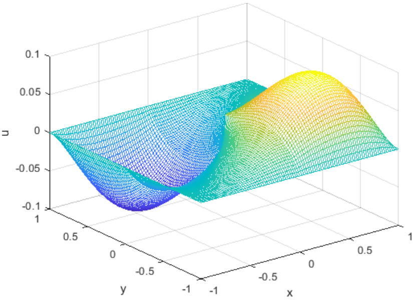


Figure – Solution u_o^2 at iteration 2, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Idea : Split the data f, g into even/odd symmetric parts, which leads to two corresponding subproblems (P_e) and (P_o). Apply the NNM to each one of these subproblems separately.

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

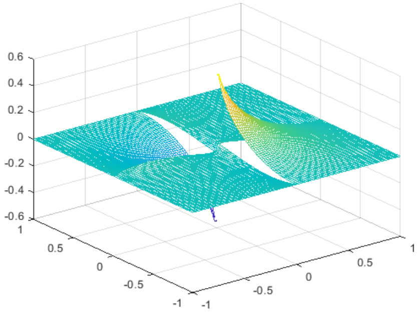


Figure – Correction ψ_o^2 at iteration 2, $\theta = 0.15$.

Convergence analysis for the NNM - Results

Let us fix an initial guess $\lambda^0 \in C^0(\Sigma)$ that is compatible with the Dirichlet boundary condition.

Proposition 1. *Taking λ_e^0 as the initial guess for the NNM applied to (P_e) produces a sequence $\left\{u_e^k\right\}_k$ that converges geometrically to the solution u_e with respect to the L^2-norm and the broken H^1-norm, for any $\theta \in (0,1)$. Moreover, the convergence factor is given by $|1-4\theta|$, which also proves that the NNM becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

Proposition 2. *Taking λ_o^0 as the initial guess for the NNM applied to (P_o) generates a sequence $\left\{u_o^k\right\}_k$ such that, for some $k_0 > 1$, the iterates u_o^k are not unique for all $k \geq k_0$. In addition, for each $k \geq k_0$, all possible u_o^k are singular at the cross-point, with a leading singularity of type $(\ln r)^{2(k-k_0)+2}$.*

$\longrightarrow$ **The method needs to be fixed for odd symmetric problems !**

Convergence analysis for the NNM - Results

Let us fix an initial guess $\lambda^0 \in C^0(\Sigma)$ that is compatible with the Dirichlet boundary condition.

Proposition 1. *Taking λ_e^0 as the initial guess for the NNM applied to (P_e) produces a sequence $\left\{u_e^k\right\}_k$ that converges geometrically to the solution u_e with respect to the L^2-norm and the broken H^1-norm, for any $\theta \in (0,1)$. Moreover, the convergence factor is given by $|1-4\theta|$, which also proves that the NNM becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

Proposition 2. *Taking λ_o^0 as the initial guess for the NNM applied to (P_o) generates a sequence $\left\{u_o^k\right\}_k$ such that, for some $k_0 > 1$, the iterates u_o^k are not unique for all $k \geq k_0$. In addition, for each $k \geq k_0$, all possible u_o^k are singular at the cross-point, with a leading singularity of type $(\ln r)^{2(k-k_0)+2}$.*

$\longrightarrow$ **The method needs to be fixed for odd symmetric problems !**

Convergence analysis for the NNM - Results

Let us fix an initial guess $\lambda^0 \in C^0(\Sigma)$ that is compatible with the Dirichlet boundary condition.

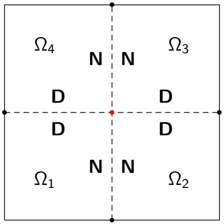
Proposition 1. *Taking λ_e^0 as the initial guess for the NNM applied to (P_e) produces a sequence $\left\{u_e^k\right\}_k$ that converges geometrically to the solution u_e with respect to the L^2-norm and the broken H^1-norm, for any $\theta \in (0,1)$. Moreover, the convergence factor is given by $|1-4\theta|$, which also proves that the NNM becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

Proposition 2. *Taking λ_o^0 as the initial guess for the NNM applied to (P_o) generates a sequence $\left\{u_o^k\right\}_k$ such that, for some $k_0 > 1$, the iterates u_o^k are not unique for all $k \geq k_0$. In addition, for each $k \geq k_0$, all possible u_o^k are singular at the cross-point, with a leading singularity of type $(\ln r)^{2(k-k_0)+2}$.*

$\longrightarrow$ **The method needs to be fixed for odd symmetric problems !**

A "mixed" Neumann-Neumann method

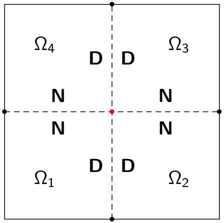
Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

A "mixed" Neumann-Neumann method

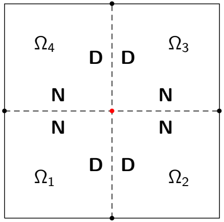
Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

A "mixed" Neumann-Neumann method

Four subdomain version : The domain Ω is split into four squared subdomains of equal area, thus involving one cross-point.



- **(Dirichlet step)** Solve a Dirichlet problem for u_i in each subdomains Ω_i.
- **(Neumann step)** Solve a Neumann problem for ψ_i in each subdomain Ω_i.
- **(Update λ)** Update the trace on each part of Σ.

Convergence analysis - Outline (odd symmetric part)

Since the method has been designed to treat specifically the odd symmetric part of the problem, we focus on convergence analysis for (P_o).

Convergence analysis for (P_o).

Find u_o solution to :

$$\begin{cases} -\Delta u_o = f_o \text{ in } \Omega \,, \\ \quad u_o = g_o \text{ on } \partial\Omega \,. \end{cases} \qquad (P_o)$$

$\longrightarrow$ At each iteration of the NNM, all local problems are well-posed.

$\longrightarrow$ Local corrections in subdomains Ω_2, Ω_3 and Ω_4 can be deduced from the local correction in Ω_1 using symmetry arguments.

$\longrightarrow$ The algorithm converges *geometrically* to the real solution u_o.

Convergence analysis - Outline (odd symmetric part)

Since the method has been designed to treat specifically the odd symmetric part of the problem, we focus on convergence analysis for (P_o).

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

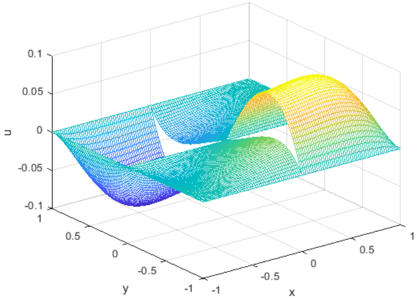


Figure – Solution u_o^1 at iteration 1, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Since the method has been designed to treat specifically the odd symmetric part of the problem, we focus on convergence analysis for (P_o).

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

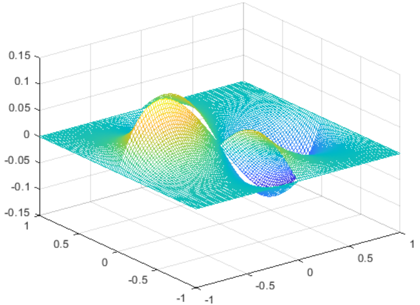


Figure – Correction ψ_o^1 at iteration 1, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Since the method has been designed to treat specifically the odd symmetric part of the problem, we focus on convergence analysis for (P_o).

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

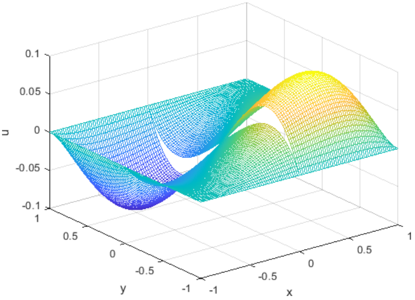


Figure – Solution u_o^2 at iteration 2, $\theta = 0.15$.

Convergence analysis - Outline (odd symmetric part)

Since the method has been designed to treat specifically the odd symmetric part of the problem, we focus on convergence analysis for (P_o).

Convergence analysis for (P_o).

Numerical illustration with $f_o = x - y$, $g_o = 0$ and $\lambda_o^0 = 0$, for $\theta = 0.15$.

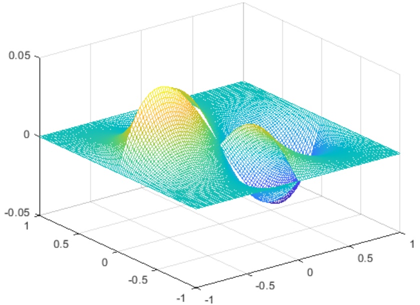


Figure – Correction ψ_o^2 at iteration 2, $\theta = 0.15$.

Convergence analysis for the mixed NNM - Results

Let us fix an initial guess $\lambda^0 \in C^0(\Sigma)$ that is compatible with the Dirichlet boundary condition.

Proposition 3. *Taking λ_o^0 as the initial guess for the mixed NNM applied to (P_o) produces a sequence $\left\{u_o^k\right\}_k$ that converges geometrically to the solution u_o with respect to the L^2-norm and the broken H^1-norm, for any $\theta \in (0, 1/2)$. Moreover, the convergence factor is given by $|1 - 4\theta|$, which also proves that the mixed NNM becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

$\longrightarrow$ **We can build a modified NNM that converges geometrically for both parts (even AND odd symmetric) !**

Convergence analysis for the mixed NNM - Results

Let us fix an initial guess $\lambda^0 \in C^0(\Sigma)$ that is compatible with the Dirichlet boundary condition.

Proposition 3. *Taking λ_o^0 as the initial guess for the mixed NNM applied to (P_o) produces a sequence $\left\{u_o^k\right\}_k$ that converges geometrically to the solution u_o with respect to the L^2-norm and the broken H^1-norm, for any $\theta \in (0, 1/2)$. Moreover, the convergence factor is given by $|1-4\theta|$, which also proves that the mixed NNM becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

$\longrightarrow$ **We can build a modified NNM that converges geometrically for both parts (even AND odd symmetric) !**

A modified Neumann-Neumann method

Here are the different steps of our modified NNM to solve (P) starting from an initial guess $\lambda^0 \in C^0(\Sigma)$ compatible with the Dirichlet boundary condition, and a relaxation parameter $\theta \in (0, 1/2)$.

1. Decompose the data into their even/odd symmetric parts in order to get (P_e) and (P_o).
2. Solve in parallel :
   - (P_e) using the standard NNM starting from λ_e^0,
   - (P_o) using the mixed NNM starting from λ_o^0.
3. Recompose the solution $u = u_e + u_o$.

Remark : It is actually enough to solve for u_e and u_o in $\Omega_1 \cup \Omega_2$, and then extend them to the whole domain Ω by symmetry.

Proposition 4. *The modified NNM applied to (P) converges geometrically for any $\theta \in (0, 1/2)$. Moreover, it becomes a direct solver when $\theta = \frac{1}{4}$.*

A modified Neumann-Neumann method

Here are the different steps of our modified NNM to solve (P) starting from an initial guess $\lambda^0 \in C^0(\Sigma)$ compatible with the Dirichlet boundary condition, and a relaxation parameter $\theta \in (0, 1/2)$.

1. Decompose the data into their even/odd symmetric parts in order to get (P_e) and (P_o).
2. Solve in parallel :
   - (P_e) using the standard NNM starting from λ_e^0,
   - (P_o) using the mixed NNM starting from λ_o^0.
3. Recompose the solution $u = u_e + u_o$.

Remark : It is actually enough to solve for u_e and u_o in $\Omega_1 \cup \Omega_2$, and then extend them to the whole domain Ω by symmetry.

Proposition 4. *The modified NNM applied to (P) converges geometrically for any $\theta \in (0, 1/2)$. Moreover, it becomes a direct solver when $\theta = \frac{1}{4}$.*

A modified Neumann-Neumann method

Here are the different steps of our modified NNM to solve (P) starting from an initial guess $\lambda^0 \in C^0(\Sigma)$ compatible with the Dirichlet boundary condition, and a relaxation parameter $\theta \in (0, 1/2)$.

1. Decompose the data into their even/odd symmetric parts in order to get (P_e) and (P_o).
2. Solve in parallel :
   - (P_e) using the standard NNM starting from λ_e^0,
   - (P_o) using the mixed NNM starting from λ_o^0.
3. Recompose the solution $u = u_e + u_o$.

Remark : It is actually enough to solve for u_e and u_o in $\Omega_1 \cup \Omega_2$, and then extend them to the whole domain Ω by symmetry.

Proposition 4. *The modified NNM applied to (P) converges geometrically for any $\theta \in (0, 1/2)$. Moreover, it becomes a direct solver when $\theta = \frac{1}{4}$.*

Even symmetric example

Data : The domain Ω is discretized using a uniform cartesian grid, with mesh size $h = 0.02$. The physical data are such that $f = 1$ in Ω and $g = 0$ on $\partial\Omega$. The initial guess is set to $\lambda^0 = 0$ on Σ.

The "exact" solution looks like :

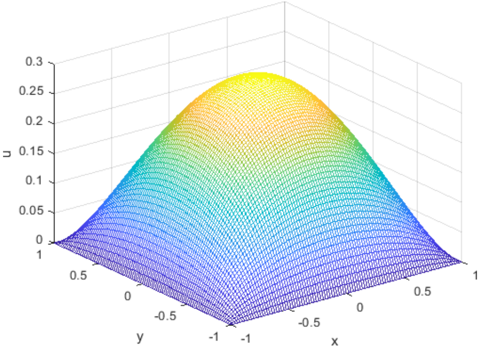


Figure – Exact solution.

Even symmetric example

Data : The domain Ω is discretized using a uniform cartesian grid, with mesh size $h = 0.02$. The physical data are such that $f = 1$ in Ω and $g = 0$ on $\partial\Omega$. The initial guess is set to $\lambda^0 = 0$ on Σ.

We apply the modified NNM with $\theta = 0.25$.

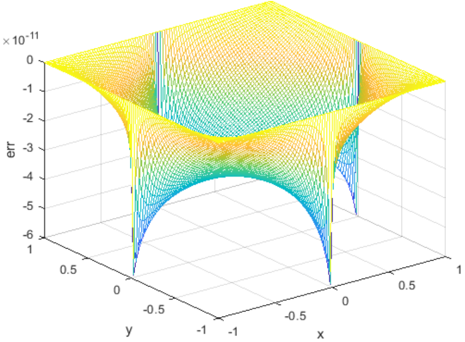


Figure – Absolute error for the modified NNM at iteration 2, $\theta = 0.25$.

Even symmetric example

Data : The domain Ω is discretized using a uniform cartesian grid, with mesh size $h = 0.02$. The physical data are such that $f = 1$ in Ω and $g = 0$ on $\partial\Omega$. The initial guess is set to $\lambda^0 = 0$ on Σ.

We apply the modified NNM with $\theta = 0.23$.

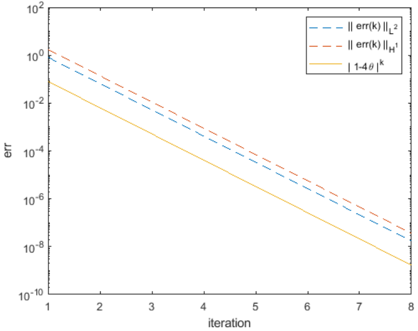


Figure – Norm of the absolute error, $\theta = 0.23$.

Odd symmetric example

Data : The domain Ω is discretized using a uniform cartesian grid, with mesh size $h = 0.02$. The physical data are such that $f = x - y$ in Ω and $g = 0$ on $\partial\Omega$. The initial guess is set to $\lambda^0 = 0$ on Σ.

The "exact" solution looks like :

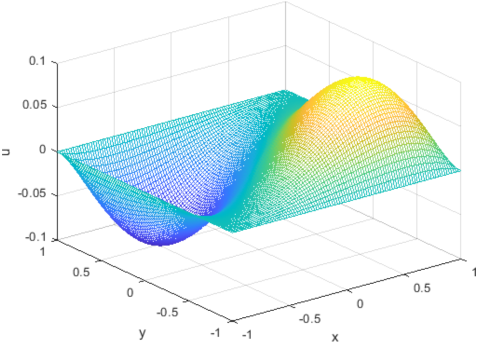


Figure – Exact solution.

Odd symmetric example

Data : The domain Ω is discretized using a uniform cartesian grid, with mesh size $h = 0.02$. The physical data are such that $f = x - y$ in Ω and $g = 0$ on $\partial\Omega$. The initial guess is set to $\lambda^0 = 0$ on Σ.

We apply the modified NNM with $\theta = 0.25$.

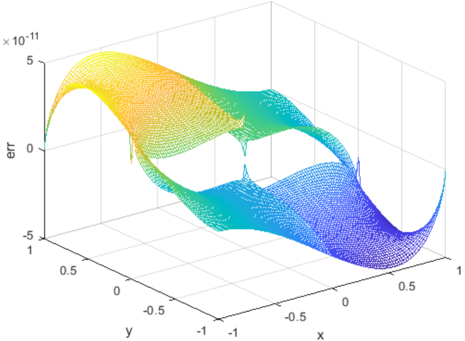


Figure – Absolute error for the modified NNM at iteration 2, $\theta = 0.25$.

Odd symmetric example

Data : The domain Ω is discretized using a uniform cartesian grid, with mesh size $h = 0.02$. The physical data are such that $f = x - y$ in Ω and $g = 0$ on $\partial\Omega$. The initial guess is set to $\lambda^0 = 0$ on Σ.

We apply the modified NNM with $\theta = 0.23$.

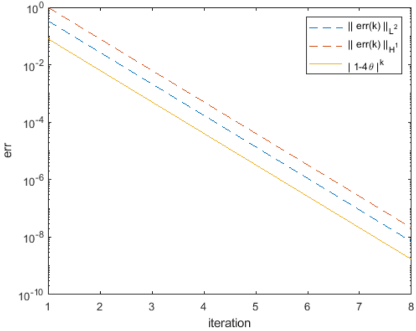


Figure – Norm of the absolute error, $\theta = 0.23$.

Conclusion and outlook

Summary :

- Study of the standard NN method for the Poisson problem in a specific configuration involving one cross-point.
- Introduction of a modified NNM that converges geometrically for any data f, g and initial guess λ^0, and that becomes a direct solver for $\theta = 1/4$.

Other results (not mentioned here) :

- Similar results for the Dirichlet-Neumann method.
- Extension to the 3D case (cube divided in 4 or 8 subdomains).

Future work :

- Extension to more general cross-points (not necessarily rectilinear, or involving a number of subdomains $N \neq 4$).
- Extension to more than one cross-points (multilevel method).

Conclusion and outlook

Summary :

- Study of the standard NN method for the Poisson problem in a specific configuration involving one cross-point.
- Introduction of a modified NNM that converges geometrically for any data f, g and initial guess λ^0, and that becomes a direct solver for $\theta = 1/4$.

Other results (not mentioned here) :

- Similar results for the Dirichlet-Neumann method.
- Extension to the 3D case (cube divided in 4 or 8 subdomains).

Future work :

- Extension to more general cross-points (not necessarily rectilinear, or involving a number of subdomains $N \neq 4$).
- Extension to more than one cross-points (multilevel method).

Conclusion and outlook

Summary :

- Study of the standard NN method for the Poisson problem in a specific configuration involving one cross-point.
- Introduction of a modified NNM that converges geometrically for any data f, g and initial guess λ^0, and that becomes a direct solver for $\theta = 1/4$.

Other results (not mentioned here) :

- Similar results for the Dirichlet-Neumann method.
- Extension to the 3D case (cube divided in 4 or 8 subdomains).

Future work :

- Extension to more general cross-points (not necessarily rectilinear, or involving a number of subdomains $N \neq 4$).
- Extension to more than one cross-points (multilevel method).

Thank you for your attention !