

Partial Cells Project

Contents

1 One-dimensional Approach	8
1.1 Rectangular Finite Element Method and Cut Cell Approach- until 3/4/2009	8
1.2 First Attempt: Underintegration	8
1.3 Second Attempt: Naive Approach	9
1.3.1 Linear Interpolation	10
1.3.2 Quadratic Interpolation	11
1.3.3 Cubic Interpolation	12
1.4 Third Attempt: Tranair Approach	13
1.4.1 With consideration of the DE contribution to the entire element	14
1.4.2 Without consideration of the DE contribution to the element	16
1.5 Finite Volume Method (Prof. Papadopoulos Approach)	17
1.5.1 One dimensional finite volume method	18
1.5.2 Results	20
1.5.3 Cut Cell	21
1.6 First Attempt: Finite Volume Method (Prof. Colella Approach)	23
1.6.1 Theory	23
1.6.2 Results	23
1.7 Second Attempt: Finite Volume Method (Prof. Colella Approach)	25
1.7.1 Theory	25
1.7.2 Results	28
1.8 Consideration of the PDE to the last element (Finite Element Method)	30
1.8.1 Results	30
2 Comparison Of All Of The Methods	31
2.1 PP Finite Volume	31
2.2 PC Finite Volume	32
2.3 Finite Element 2 Elements Coupling	33
2.4 Finite Element 3 Elements Coupling	34
2.5 Finite Element Tranair Approach With Consideration of the DE Contribution	35
2.5.1 Penalty Method	35
2.5.2 Lagrange Multiplier	36
2.6 Remarks	36
3 Comparisons: New Attempt	36
3.1 PP Finite Volume	37
3.2 Scaled PP Finite Volume	37
3.3 PC Finite Volume	38
3.4 Scaled PC Finite Volume	38

3.5 Finite Element 2 Elements Coupling With PDE contribution	39
3.6 Finite Element Tranair Approach	40
4 Comparison Study	41
4.1 PP Finite Volume	41
4.2 PC Finite Volume	42
4.3 Finite Element 2 Elements Coupling With Full PDE contribution	43
4.4 Finite Element 2 Elements Coupling With Scaled PDE contribution	44
4.5 Finite Element Tranair Approach	46
4.6 Comments	48
4.7 Finite Element 2 Elements Coupling With Full PDE contribution Linear BC Interpolation	49
5 New Problem : Quadratic Solution	50
6 Neumann Boundary Conditions	53
6.1 Derivative of Quadratic Interpolation: Bug in analytical solution	53
6.2 Derivative of Linear Interpolation: Bug in analytical solution	55
6.3 Corrected: Derivative of Linear Interpolation and Quadratic Interpolation .	57
6.3.1 Derivative of Linear Interpolation	57
6.3.2 Derivative of Quadratic Interpolation	59
6.3.3 Interpretations	61
6.4 Finite Element Flux Solution	61
7 Two-Dimensional Approach	67
7.1 Canonical Problem: Circular Physical Domain	67
7.2 Analytical Solution	67
7.3 Weak Form	70
7.4 Discretization	71
7.5 Geometry	71
7.6 Solution Method	73
7.7 Convergence Study: First Attempt	77
7.8 Comparison to the One-Dimensional Algorithm	81
7.8.1 Eigenvalue-Analysis for the Quadratic Interpolation Case	84
7.8.2 Different Error Norms and Independent Variables	88
7.9 Different Geometry: Rectangular Domain	93
7.9.1 Analytical Solution: Wrong One	94
7.9.2 Convergence Analysis	94
7.9.3 Analytical Solution: Corrected	95
8 Circular Domain: Convergence Study the 2nd	96
8.1 New Independent Variable: Element Width	97
8.2 Bidirectional Interpolation vs. Simple Interpolation	99

9 Structured vs. Unstructured Meshing	101
9.1 Unstructured Mesh	102
9.2 Improving Mesh Quality: Neighbor Averaging	103
9.3 Improving Mesh Quality: Poisson Smoothing	104
9.4 Comparison of Convergence Rates: Original Mesh	104
9.5 Comparison of Convergence Rates: Neighbor Averaged Mesh	106
10 Torus: Homogeneous Dirichlet Boundary Conditions	108
10.1 Analytical Solution	109
10.2 Any Modifications to the Algorithm Necessary?	110
10.3 Results	110
10.3.1 Convergence in H_0 -norm	112
10.3.2 Convergence in H_1 -norm	112
10.3.3 Convergence in Energy-Norm	113
11 Torus: Homogeneous Neumann Boundary Conditions	113
11.0.4 Analytical Solution	115
11.1 Debugging	115
11.1.1 Suspection	118
11.1.2 Zero Neumann Boundary Conditions	119
11.1.3 Flipping the Neumann and the Dirichlet Boundary Sides	119
11.1.4 Circle: Homogeneous Neumann Boundary Conditions and Dirichlet at One Single Point	120
11.1.5 Dirichlet-Dirichlet-Problem as Input	120
11.2 Working Version	120
11.2.1 Changes to the Algorithm	121
11.2.2 Results	122
11.2.3 Convergence Behavior	123
11.2.4 Dirichlet-Dirichlet-Problem with Finite Volumes and Partial Cells 1d	124
11.2.5 Dirichlet-Neuman-Problem with Finite Volumes and Partial Cells 1d	126
11.2.6 Dirichlet-Dirichlet-Problem with Finite Elements and Partial Cells 1d	128
11.2.7 Dirichlet-Neumann-Problem with Finite Elements and Partial Cells 1d	130
11.2.8 Dirichlet-Neumann-Problem with Finite Elements without partial cells 2d	132
11.2.9 Length of Partial Physical Domain and Convergence Order (1d, Dirichlet Boundary Condition)	
11.2.10 Length of Partial Cell and Convergence Order (1d, Neumann Boundary Conditions)	139
11.2.11 Least Squares: Different Method for Imposing Neumann Boundary Conditions	145
12 Complex Shaped Physical Domain	146

List of Tables

1	Convergence Test Results	78
2	Additional Convergence Test Results	79
3	Convergence Test Results of One-dimensional Case Without Coupling	81
4	Convergence Test Results With Coupling	83
5	Comparison of Analytical and Numerical Solution for $h = 0.5$	83
6	Comparison of Analytical and Numerical Solution for $h = 0.25$	83
7	Comparison of Analytical and Numerical Solution for $h = 0.125$	84
8	Condition Number of the System Matrix For Different Element Sizes Before Imposing Boundary C	
9	Rectangular Domain: Convergence Test Results	95

List of Figures

1	Convergence Study for regular FEM vs. Underintegrated Cut Cell	9
2	Linear Interpolation (constraint on the last element)	10
3	Quadratic Interpolation (constraint on the last two elements)	11
4	Result of Quadratic Interpolation	12
5	Cubic Interpolation (constraint on the last three elements)	13
6	Regular FEM Solution	14
7	Cut Cell Solution for different weight factors 1,5 ,10, 100 and 1000	15
8	Effect of weight factor on the solution	16
9	Effect of mesh size on the solution for different weight factors	16
10	Without considering the contribution of the DE to the last element	17
11	Domain discretization	18
12	Ghost centroid	19
13	Solution	21
14	Solution With Cut Cell for different values of α	23
15	Convergence Study for $\alpha = 0.05$	24
16	Solution With Cut Cell for different values of α	25
17	Dependency of the Derivative at the Vertex on the Centroids Values	26
18	Solution for different values of α	29
19	Convergence Study for $\alpha = 0.8$	30
20	Finite Element with Consideration of PDE	31
21	Position of the unknowns: Finite Volume Method (upper), Finite Element Method (bottom)	36
22	Solution with Finite Volume PP	42
23	Solution with Finite Volume PC	43
24	Solution with Full PDE Contribution	44
25	Solution with Partial PDE Contribution	46
26	Solution with Tranair Approach	47
27	Solution with Full PDE Contribution Linear BC Interpolation	50
28	Quadratic Solution of Finite Volume Method	51
29	Quadratic Solution of Finite Element Method With Linear Interpolation of Boundary Conditions	52
30	Quadratic Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions	
31	Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions)	
32	Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions)	
33	Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions)	
34	Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions)	
35	Flux Solution (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions)	
36	Flux Solution (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions)	
37	Flux Solution (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions)	
38	Flux Solution (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions)	
39	Circular Physical Domain and Cartersian Grid	72
40	3 Types of elements: partial (green), full (blue), outside of domain (red) . . .	73

41	Partial Element Cases: (a) 3 hanging nodes and 2 intersection points (creation of a “fake” intersection)	
42	Special Case of Elements Outside of Domain	76
43	One Element Rectangular Grid	77
44	Convergence Study	78
45	Corrected Convergence Study	79
46	Analytical Solution for Circular Domain	80
47	Numerical Solution for Circular Domain	80
48	Convergence Study of One-dimensional Case Without Coupling	82
49	Correlation of the error and the condition number of the system	85
50	Correlation of the error and the condition number of the system vs. Round-off Error Upper Bound	87
51	Neumann Problem: Derivative of Linear Interpolation vs. Derivative of Quadratic Interpolation	87
52	Dirichlet Problem: Linear Interpolation vs. Quadratic Interpolation	88
53	H ₀ error norm vs. the number of interior nodes	89
54	H ₀ error norm vs. the number of elements in one-direction	90
55	H ₁ error norm vs. the number of interior nodes	91
56	H ₁ error norm vs. the number of elements in one-direction (the xlabel should read ‘the number of el’)	
57	Energy error norm vs. the number of interior nodes	92
58	Energy error norm vs. the number of elements in one-direction	93
59	Correlation between the number of interior nodes and the number of elements per direction	93
60	Rectangular Domain: Convergence Study	95
61	Rectangular Domain: Convergence Study	96
62	Each Intersection Point Involves Two Element Nodes	97
63	Convergence rate in the H ₀ -error norm	98
64	Convergence rate in the H ₁ -error norm	98
65	Convergence rate in the energy norm	99
66	Convergence rate for bidirectional and simple interpolation in the H ₀ -error norm	100
67	Convergence rate for bidirectional and simple interpolation in the H ₁ -error norm	100
68	Convergence rate for bidirectional and simple interpolation in the energy error norm	101
69	Starting Discretization of the Circle	102
70	Sequence of Mesh Refinements of the Circle	102
71	Meshed Circle Before and After Neighbor Averaging (40 times)	103
72	Convergence of Structured vs. Unstructured Mesh in the H ₀ -Error Norm	104
73	Convergence of Structured vs. Unstructured Mesh in the H ₁ -Error Norm	105
74	Convergence of Structured vs. Unstructured Mesh in the Energy-Error Norm	105
75	Convergence of Initial vs. Improved Mesh in the H ₀ -Error Norm	106
76	Convergence of Initial vs. Improved Mesh in the H ₁ -Error Norm	107
77	Convergence of Initial vs. Improved Mesh in the Energy Error Norm	107
78	Torus Geometry	108
79	Numerical(+) vs. Analytical (o) Solution	111
80	Convergence in the H_0 -Norm	112
81	Convergence in the H_1 -Norm	112
82	Convergence in the Energy-Norm	113

83	Contour Plots of the Numerical Solution (left) vs. the Analytical Solution (right) for the Same Spatial Resolution	117
84	Numerical Solution for the Spatial Resolutions 128x128 and 256x256	117
85	Divergence of the Numerical Solution	118
86	Critical Case	118
87	Contour Plots of the Numerical Solution (left) vs. the Analytical Solution (right) for the Same Spatial Resolution	123
88	Convergence Behavior in the H0- error norm as a function of the grid cell size.	123
89	Convergence Behavior in the H1- error norm as a function of the grid cell size.	124
90	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Volumes and Partial Cells: H0-conforming	124
91	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Volumes and Partial Cells: H1-conforming	124
92	One-dimensional Dirichlet-Neumann-Problem solved with Finite Volumes and Partial Cells: H0-conforming	124
93	One-dimensional Dirichlet-Neumann-Problem solved with Finite Volumes and Partial Cells: H1-conforming	124
94	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
95	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
96	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
97	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
98	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
99	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
100	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
101	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
102	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
103	One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
104	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
105	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
106	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
107	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-conforming	124
108	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H0-conforming	124
109	One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-conforming	124

This document summarizes the progress of the research project based on the weekly meetings with Prof. Papadopoulos. This document was started on March the 4th 2009 and will be updated every week.

1 One-dimensional Approach

1.1 Rectangular Finite Element Method and Cut Cell Approach- until 3/4/2009

We have been working on the understanding of the cut cell approach from the finite element point of view. We started with a one-dimensional static linear elasticity problem. Note that we are using linear A one dimensional bar with a length L fixed on one under body forces of magnitude K .

$$Au_{,xx} + K = 0, \quad (1)$$

$$u_{x=0} = 0, \quad (2)$$

$$0 < x < L. \quad (3)$$

Let the discretization of the domain be extended, such that a node trespasses the actual end of the domain, i.e. the position of a node would be greater than L . Thus, the last element, from now on the cut cell, would consist of the actual bar and an empty zone.

The questions are: How would you proceed in considering the last element? What if the actual end is subject to displacement or flux boundary conditions?

1.2 First Attempt: Underintegration

With a free end, the finite element treatment of the entire domain remains the same except the last element, where the stiffness and the force contributions are integrated using the quadrature points that lie in the actual domain of the body. Quadrature points that lie on the region of the element that is not covered by the body are not considered. This method converges towards the original solution when the mesh size decreases. See Figure 1.

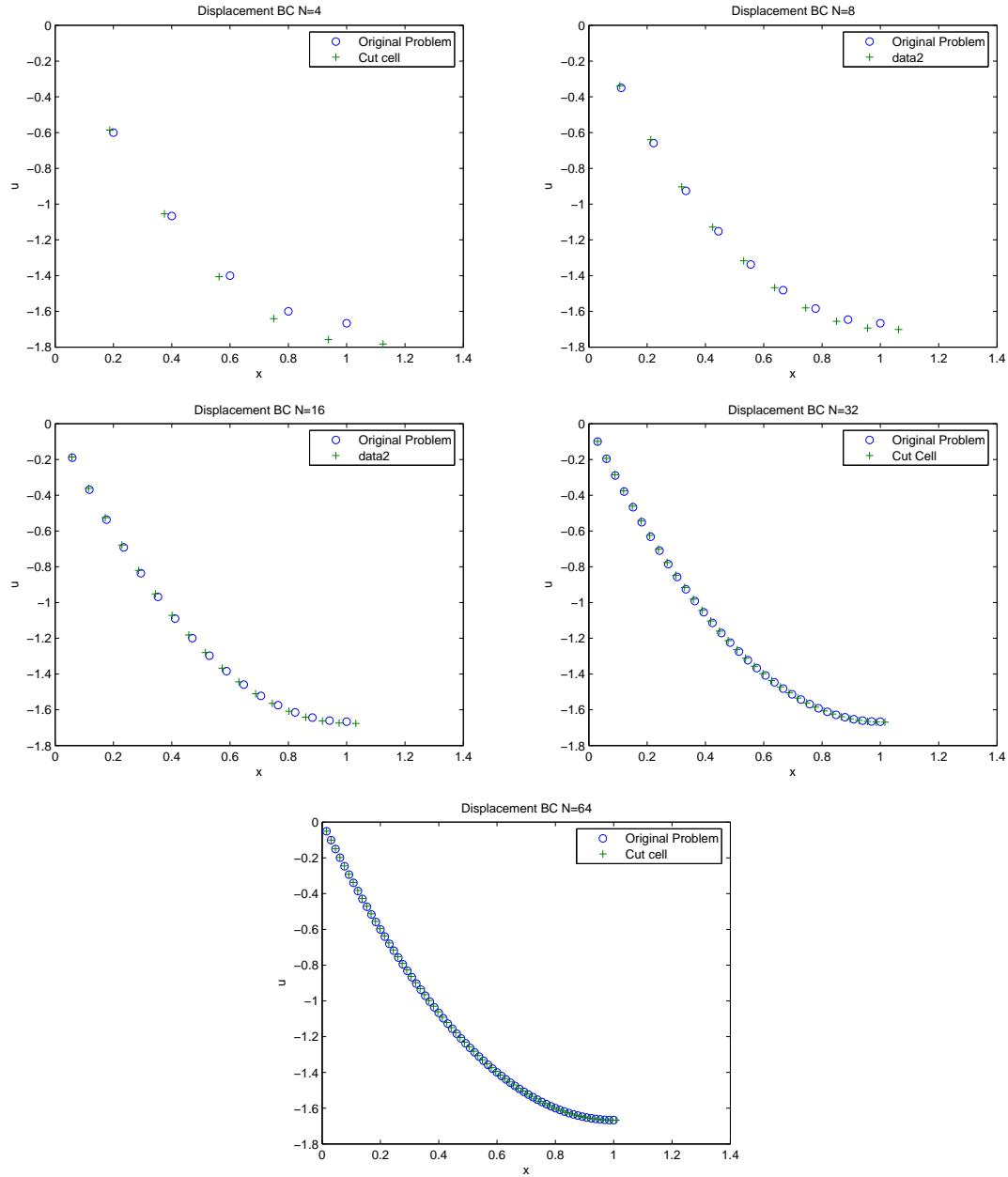


Figure 1: Convergence Study for regular FEM vs. Underintegrated Cut Cell

The same problem has been solved, when the actual end of the domain is subject to traction boundary conditions. The other end is still fixed.

1.3 Second Attempt: Naive Approach

Obviously, the problem lies in how to impose the boundary condition, such that it is not associated with an unknown locator (node). A naive idea is to interpolation between some of

the nodes, evaluate the intertoplation function at the location of the boundary condition, and obtain a constraint from that, that adds to the total system that needs to be solved.

1.3.1 Linear Interpolation

The actual end of the domain lies naturally, due to the choice of the discretization, in the last element. Since we are taking linear elements here, the end of the physical domain lies between two nodes. The solution at the last element is therefore linear, and can be written as (see also figure 2):

$$u = N_{n-1}(x)u_{n-1} + N_n(x)u_n, \quad (4)$$

$$N_{n-1}(x) = -\frac{(x - \Delta x)}{\Delta x}, \quad (5)$$

$$N_n(x) = \frac{x}{\Delta x}, \quad (6)$$

where u_j is the displacement at node j and Δx is the size of the element.

Imposing the boundary condition, $u|_L = \bar{u}$, assuming that the physical domain covers $\alpha\Delta x$ of the element, we obtain the following equation:

$$u = \bar{u} = N_{n-1}(\alpha\Delta x)u_{n-1} + N_n(\alpha\Delta x)u_n. \quad (7)$$

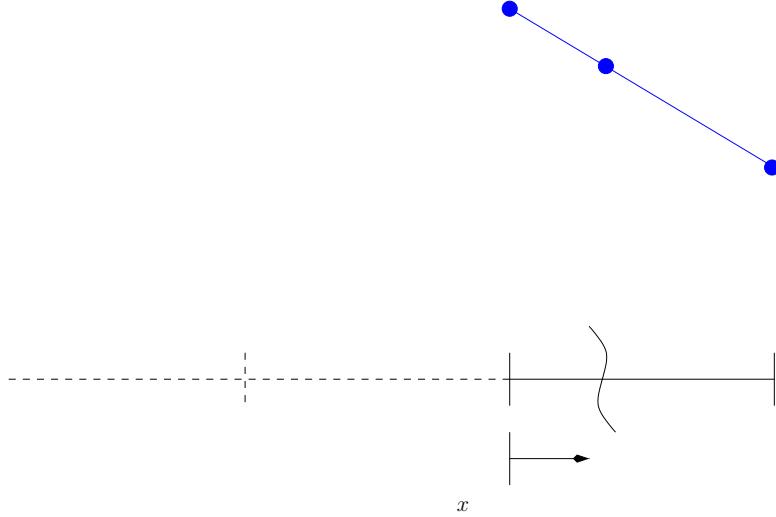


Figure 2: Linear Interpolation (constraint on the last element)

Note here, that the addition of this constraint does not account for a material property of the physical domain in the last element. Note here, that the last element does not get a contribution from the usual finite element setting.

In the case, where the body forces K do vanish, the solution is not acceptable. The last element behaves like a foam, satisfies the imposed boundary condition, but does not propagate the information to the neighbor elements about its deformation. The neighbor elements are

stiff compared to the last element and do not deform. We thought, that the use of one single element in the interpolation is not enough to forward the information to the entire domain. So, we tried to do the interpolation over the two last elements instead of one.

1.3.2 Quadratic Interpolation

The interpolation is done over the two last elements (see figure 3) and reads now:

$$u = N_{n-2}u_{n-2} + N_{n-1}(x)u_{n-1} + N_n(x)u_n, \quad (8)$$

$$N_{n-2}(x) = \frac{(x - \Delta x)(x - 2\Delta x)}{2\Delta x^2}, \quad (9)$$

$$N_{n-1}(x) = -\frac{x(x - 2\Delta x)}{\Delta x^2}, \quad (10)$$

$$N_n(x) = \frac{x(x - \Delta x)}{2\Delta x^2}. \quad (11)$$

$$(12)$$

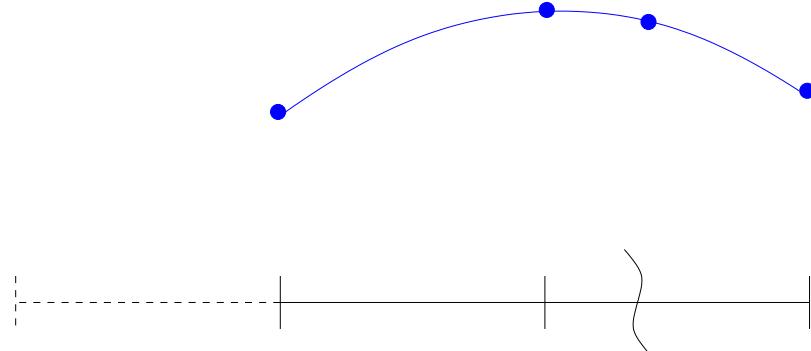


Figure 3: Quadratic Interpolation (constraint on the last two elements)

We noticed the same behavior despite the use of a higher order interpolation. The information is not spread to the neighbor elements, and hence do not deform, due to the large difference in the material properties. The results are depicted in figure:

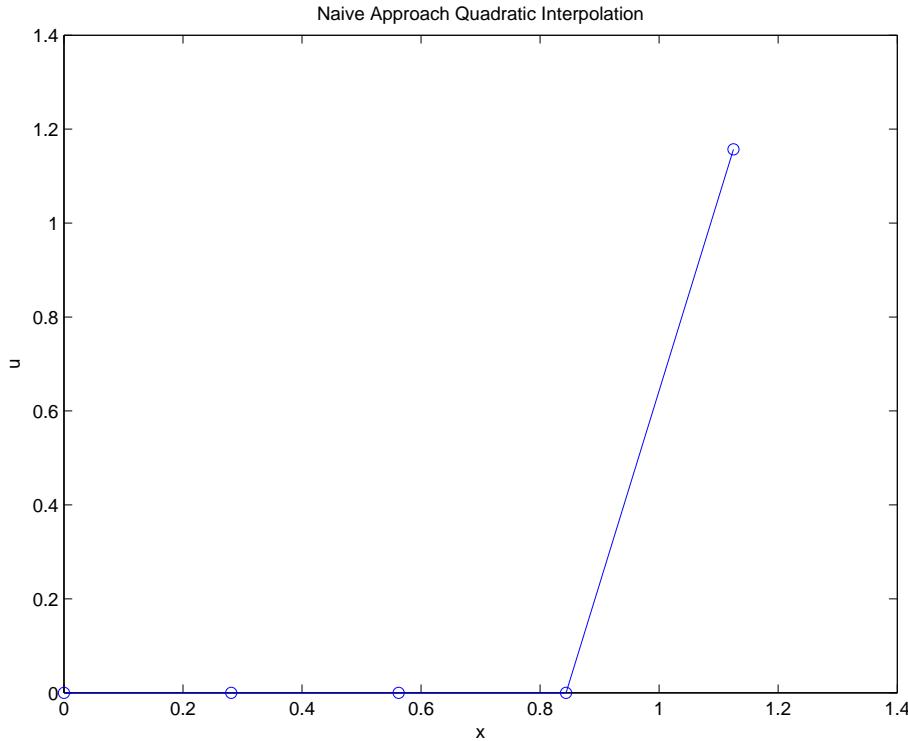


Figure 4: Result of Quadratic Interpolation

1.3.3 Cubic Interpolation

The information from three neighbor elements is used to interpolate the value of the displacement boundary condition.

$$u = N_{n-3}u_{n-3} + N_{n-2}u_{n-2} + N_{n-1}(x)u_{n-1} + N_n(x)u_n, \quad (13)$$

$$N_{n-3}(x) = -\frac{(x - \Delta x)(x - 2\Delta x)(x - 3\Delta x)}{6\Delta x^3} \quad (14)$$

$$N_{n-2}(x) = \frac{x(x - 2\Delta x)(x - 3\Delta x)}{2\Delta x^3}, \quad (15)$$

$$N_{n-1}(x) = -\frac{x(x - \Delta x)(x - 3\Delta x)}{2\Delta x^3}, \quad (16)$$

$$N_n(x) = \frac{x(x - \Delta x)(x - 2\Delta x)}{6\Delta x^3}. \quad (17)$$

$$(18)$$

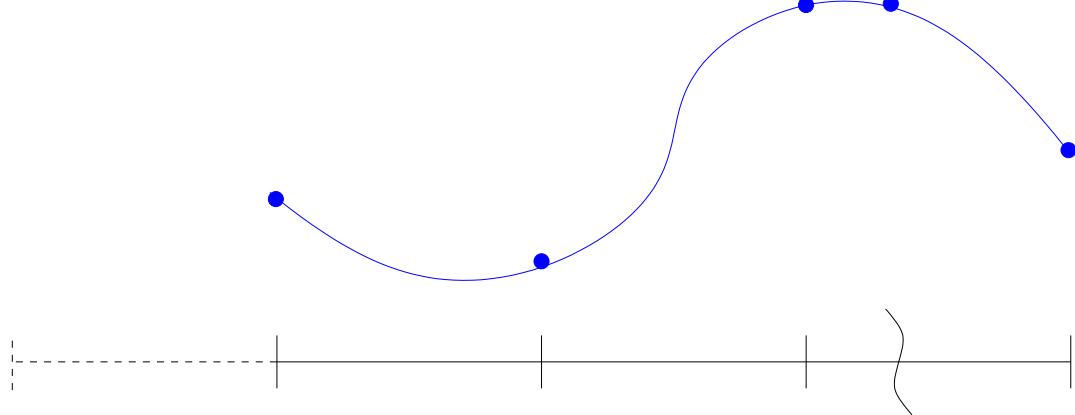


Figure 5: Cubic Interpolation (constraint on the last three elements)

The cubic interpolation did not help either, and we obtain the same behavior.

1.4 Third Attempt: Tranair Approach

The weak form of the problem given in (1) reads:

$$Au_{,x}\xi|_0^L - \int Au_{,x}\xi_{,x} dx + \int K\xi dx = 0,$$

where ξ is the testfunction. Since we do not have any traction boundary conditions prescribed, the weak form simplifies to:

$$-\int Au_{,x}\xi_{,x} dx + \int K\xi dx = 0.$$

We add the following term, to enforce the Dirichlet boundary conditions:

$$-\int Au_{,x}\xi_{,x} dx + \int K\xi dx + (\xi(u - \bar{u}))|_L = 0.$$

The stiffness matrix reads:

$$\mathbf{K} = -\int Au_{,x}\xi_{,x} dx + (\xi u)|_L.$$

The force vector reads:

$$\mathbf{f} = -\int K\xi dx + \xi|_L(\bar{u}).$$

Alternatively, we solve using:

$$\begin{aligned} \mathbf{K} &= \int Au_{,x}\xi_{,x} dx - (\xi u)|_L, \\ \mathbf{f} &= \int K\xi dx - \xi|_L(\bar{u}), \end{aligned}$$

in order to obtain positive definiteness of the stiffness matrix. After discretization, the stiffness matrix reads:

$$\mathbf{K} = - \int AN_{J,x}N_{I,x} dx + (N_I N_J)|_L.$$

The force vector reads:

$$\mathbf{f} = - \int KN_I dx + N_I|_L(\bar{u}).$$

Consistently considering the signs for positive definiteness, the discretized entries read:

$$\begin{aligned}\mathbf{K} &= \int AN_{J,x}N_{I,x} dx - (N_I N_J)|_L, \\ \mathbf{f} &= \int KN_I dx - N_I|_L(\bar{u}).\end{aligned}$$

The regular FEM solution to the above problem with a Dirichlet boundary condition $u(x = L) = 0.5$ is depicted in figure 6.

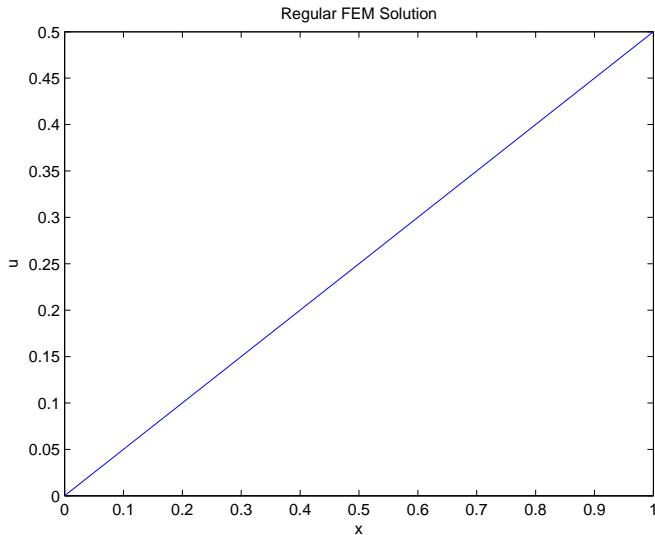


Figure 6: Regular FEM Solution

1.4.1 With consideration of the DE contribution to the entire element

For the cut cell element, only the differential equation resolved over the part of the element that does belong to the physical domain is considered. Hence the multiplication with the ratio of the physical domain to the total element length α . If we consider the differential equation contribution to the last element, and we weight the contributions to the stiffness and the load vector with a penalty parameter P^* , i.e.

$$\mathbf{K} = -\alpha \int AN_{J,x}N_{I,x} dx + P^*(N_I N_J)|_L,$$

and

$$\mathbf{f} = -\alpha \int K N_I dx + P^* N_I|_L(\bar{u}),$$

and if we consistently consider the signs for positive definiteness

$$\begin{aligned}\mathbf{K} &= \alpha \int A N_{J,x} N_{I,x} dx - P^*(N_I N_J)|_L, \\ \mathbf{f} &= \alpha \int K N_I dx - P^* N_I|_L(\bar{u}),\end{aligned}$$

we obtain the results presented in 7:

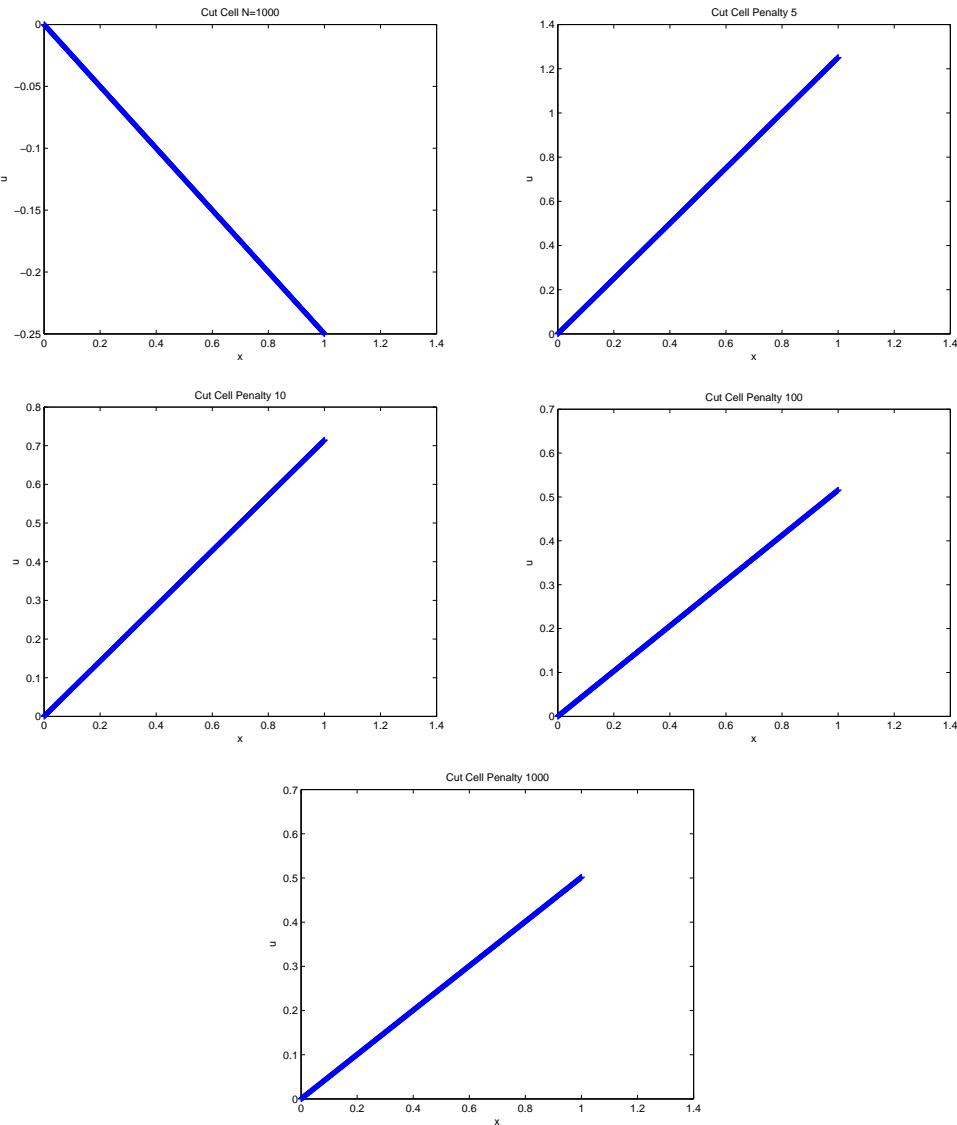


Figure 7: Cut Cell Solution for different weight factors 1 ,5 ,10, 100 and 1000

The summary of the effect of the weight parameter on the deflection of the end tip of the one-dimensional domain is given in figure 8:

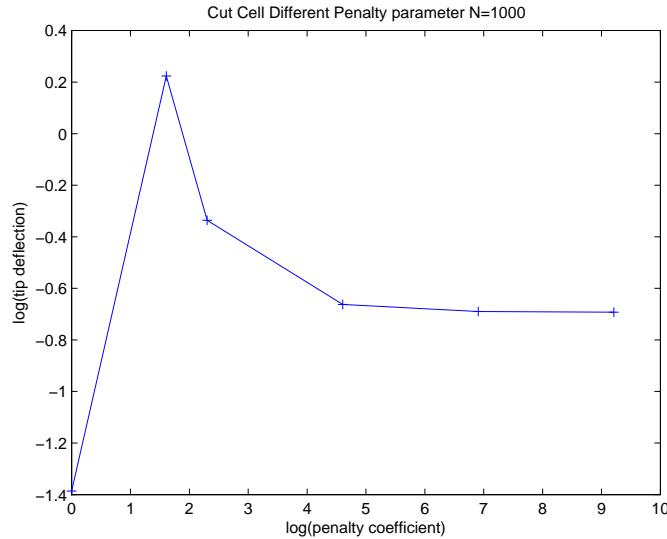


Figure 8: Effect of weight factor on the solution

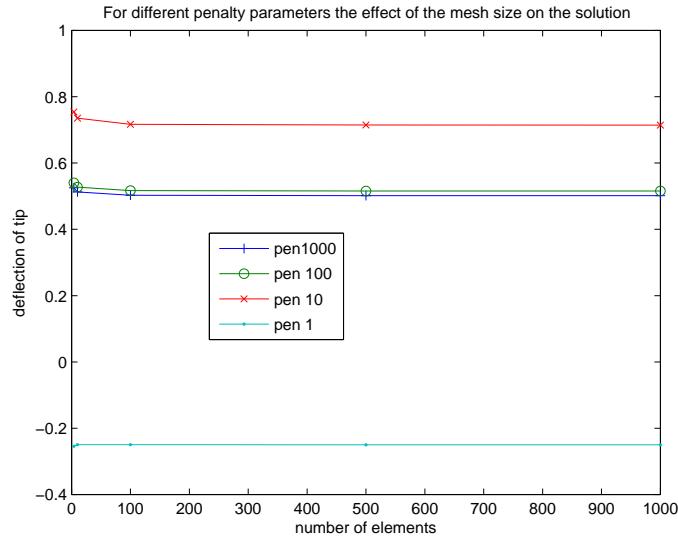


Figure 9: Effect of mesh size on the solution for different weight factors

1.4.2 Without consideration of the DE contribution to the element

If we do not consider the contributions from the differential equation to the last element, i.e.

$$\mathbf{K} = P^*(N_I N_J)|_L,$$

and

$$\mathbf{f} = P^* N_I|_L(\bar{u}),$$

we return to the naive approach we tried before (section 1.3). The results are depicted in figure 10.

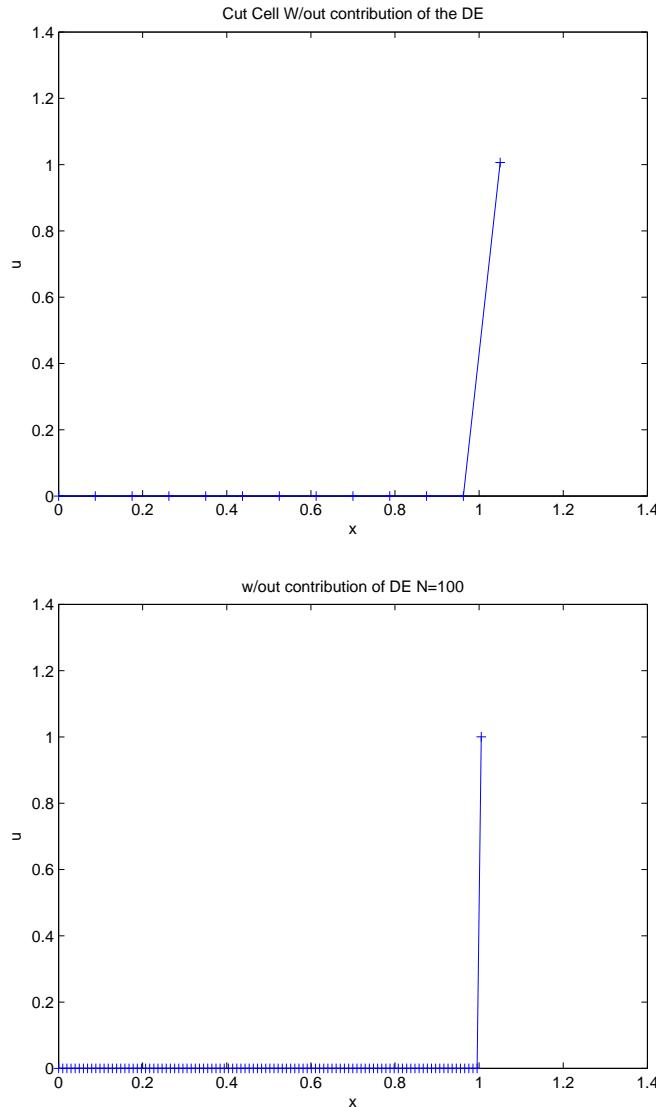


Figure 10: Without considering the contribution of the DE to the last element

1.5 Finite Volume Method (Prof. Papadopoulos Approach)

As the above described attempts have failed, we wanted to understand the treatment of the cut cells in the finite volume method, and map the similarities to the finite element method,

to figure out the correct method.¹

1.5.1 One dimensional finite volume method

We want to solve the same problem given in (1). The system is discretized into n control volumes $[x_i, x_{i+1}]$. Each control volume is assigned a centroid $x_{i+\frac{1}{2}}$ defined by:

$$x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2}. \quad (19)$$

See Figure 11.

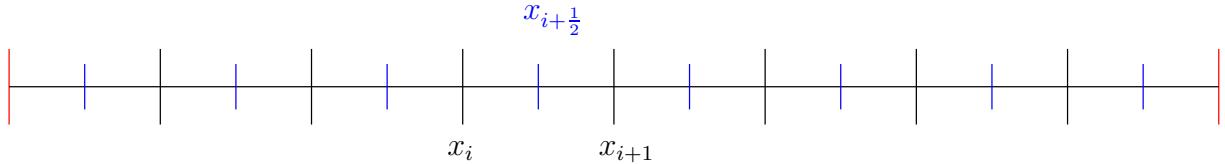


Figure 11: Domain discretization

We consider Dirichlet boundary conditions on both ends:

$$u(0) = \bar{u}_0, \quad (20)$$

$$u(L) = \bar{u}_L. \quad (21)$$

The problem can be considered in every cell, s.t.

$$Au_{,xx} = \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} Au_{,xx} dx, \quad (22)$$

$$\Delta x = x_{i+1} - x_i. \quad (23)$$

Using the mid-point rule, the above expression can be approximated by:

$$Au_{,xx} = \frac{1}{\Delta x} \Delta x Au_{,xx}|_{x_{i+\frac{1}{2}}}, \quad (24)$$

$$\simeq Au_{,xx}|_{x_{i+\frac{1}{2}}}. \quad (25)$$

Hence, the problem is solved locally in each volume element:

$$Au_{,xx}|_{x_{i+\frac{1}{2}}} = -K. \quad (26)$$

¹This approach has been derived by Prof. Papadopolous. According to Prof. Colella, this approach corresponds to the Shortley-Weller scheme, published in 1938.

The second order derivative at the centroids is approximated using a Taylor series expansion, assuming that the remainder term vanishes as the control volume size becomes smaller and smaller. We have:

$$u(x_{i+\frac{3}{2}}) = u(x_{i+\frac{1}{2}}) + u_{,x}|_{x_{i+\frac{1}{2}}} + \frac{\Delta x^2}{2}u_{,xx}|_{x_{i+\frac{1}{2}}} + \frac{\Delta x^3}{6}u_{,xxx}|_{x_{i+\frac{1}{2}}} + \frac{\Delta x^4}{24}u_{,xxxx}|_{x_i+\xi_1\Delta x}, \quad (27)$$

$$u(x_{i-\frac{1}{2}}) = u(x_{i+\frac{1}{2}}) - u_{,x}|_{x_{i+\frac{1}{2}}} + \frac{\Delta x^2}{2}u_{,xx}|_{x_{i+\frac{1}{2}}} - \frac{\Delta x^3}{6}u_{,xxx}|_{x_{i+\frac{1}{2}}} + \frac{\Delta x^4}{24}u_{,xxxx}|_{x_i+\xi_2\Delta x}. \quad (28)$$

Adding these last equations, we obtain:

$$u_{,xx}|_{x_{i+\frac{1}{2}}} = \frac{u(x_{i+\frac{3}{2}}) + u(x_{i-\frac{1}{2}}) - 2u(x_{i+\frac{1}{2}})}{\Delta x^2}, \quad (29)$$

with

$$\frac{\Delta x^4}{24}u_{,xxxx}|_{x_i+\xi_1\Delta x} \rightarrow 0, \quad (30)$$

$$\frac{\Delta x^4}{24}u_{,xxxx}|_{x_i+\xi_2\Delta x} \rightarrow 0. \quad (31)$$

We obtain equations for the centroids of the inner volumes:

$$A \frac{u(x_{i+\frac{3}{2}}) + u(x_{i-\frac{1}{2}}) - 2u(x_{i+\frac{1}{2}})}{\Delta x^2} = -K, \quad (32)$$

$$u(x_{i+\frac{3}{2}}) + u(x_{i-\frac{1}{2}}) - 2u(x_{i+\frac{1}{2}}) = \frac{-K\Delta x^2}{A}. \quad (33)$$

Clearly, this is not the case for the boundary volume elements. The way we proceed is to build a ghost centroid adjacent to the centroid we are looking at, as depicted in figure 12.

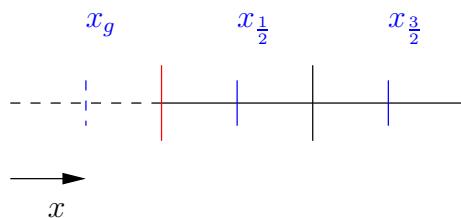


Figure 12: Ghost centroid

However, we do not want to introduce new unknowns through the ghost centroids. The solution at the ghost centroids is defined by the Dirichlet boundary conditions. The equation

on the left boundary volume centroid reads:

$$u(\Delta x) = N_1(\Delta x)u_g + N_2(\Delta x)u_{\frac{1}{2}} + N_3u_{\frac{3}{2}}, \quad (34)$$

$$= \bar{u}_0, \quad (35)$$

$$N_1(x) = \frac{(x - \frac{3\Delta x}{2})(x - \frac{5\Delta x}{2})}{2\Delta x^2}, \quad (36)$$

$$N_1(\Delta x) = \frac{3}{8}, \quad (37)$$

$$N_2(x) = -\frac{(x - \frac{\Delta x}{2})(x - \frac{5\Delta x}{2})}{\Delta x^2}, \quad (38)$$

$$N_2(\Delta x) = \frac{3}{4}, \quad (39)$$

$$N_3(x) = \frac{(x - \frac{\Delta x}{2})(x - \frac{3\Delta x}{2})}{2\Delta x^2}, \quad (40)$$

$$N_3(\Delta x) = -\frac{1}{8}. \quad (41)$$

At the end, we obtain:

$$u_g = \frac{8}{3}(\bar{u}_0 - \frac{3}{4}u_{\frac{1}{2}} + \frac{1}{8}u_{\frac{3}{2}}), \quad (42)$$

$$-4u_{\frac{1}{2}} + \frac{4}{3}u_{\frac{3}{2}} = \frac{-K\Delta x^2}{A} - \frac{8}{3}\bar{u}_0. \quad (43)$$

Analogously, we obtain an equation for the right boundary volume centroid:

$$u_{gL} = \frac{8}{3}(\bar{u}_L - \frac{3}{4}u_{n-\frac{3}{2}} + \frac{1}{8}u_{n-\frac{1}{2}}), \quad (44)$$

$$-4u_{n-\frac{1}{2}} + \frac{4}{3}u_{n-\frac{3}{2}} = \frac{-K\Delta x^2}{A} - \frac{8}{3}\bar{u}_L, \quad (45)$$

with n the total number of vertices. Finally, the system of equations reads:

$$-4u_{\frac{1}{2}} + \frac{4}{3}u_{\frac{3}{2}} = \frac{-K\Delta x^2}{A} - \frac{8}{3}\bar{u}_0, \quad (46)$$

$$u(x_{i+\frac{3}{2}}) + u(x_{i-\frac{1}{2}}) - 2u(x_{i+\frac{1}{2}}) = \frac{-K\Delta x^2}{A}, \quad i = 1, n-2 \quad (47)$$

$$-4u_{n-\frac{1}{2}} + \frac{4}{3}u_{n-\frac{3}{2}} = \frac{-K\Delta x^2}{A} - \frac{8}{3}\bar{u}_L. \quad (48)$$

1.5.2 Results

The system has been solved with $A = 3$, $K = 0$, $L = 1$. $\bar{u}_0 = 0$ and $\bar{u}_L = 0.5$. The solution is depicted in figure:

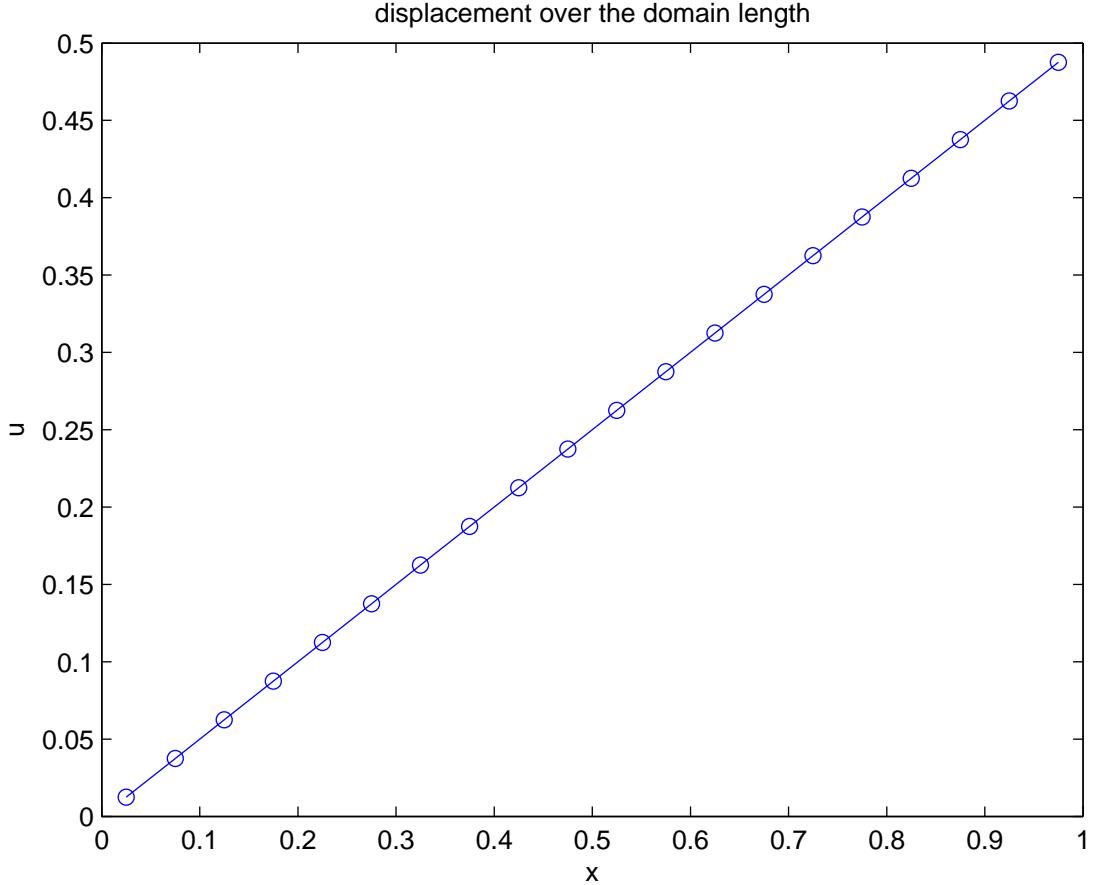


Figure 13: Solution

1.5.3 Cut Cell

In the case where the last control volume is not fully covered by the physical domain, the expressions for last system equation changes depending on the length ratio of the physical domain with respect to the full cell α . The above given expressions correspond to the special case, where $\alpha = 1$. The equation for the right boundary volume centroid reads now with:

$$u_{gL} = \frac{1}{N_3}(\bar{u}_L - N_1 \Delta x)u_{n-\frac{3}{2}} - N_2 \Delta x u_{n-\frac{1}{2}}, \quad (49)$$

$$-(2 + \frac{N_2}{N_3})u_{n-\frac{1}{2}} + (1 - \frac{N_2}{N_3})u_{n-\frac{3}{2}} = \frac{-K \Delta x^2}{A} - \frac{1}{N_3}\bar{u}_L, \quad (50)$$

$$N_1 = N_1((1 + \alpha)\Delta x), \quad (51)$$

$$N_2 = N_2((1 + \alpha)\Delta x), \quad (52)$$

$$N_3 = N_3((1 + \alpha)\Delta x). \quad (53)$$

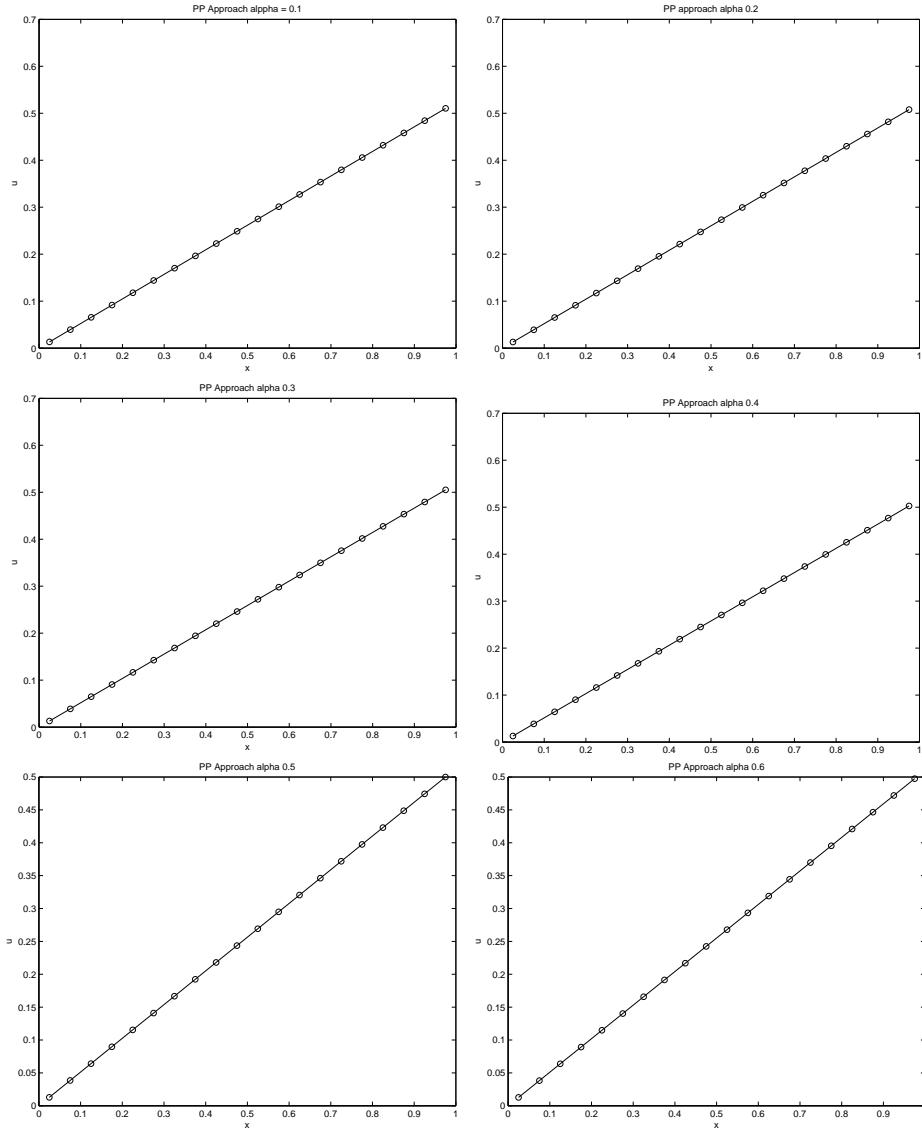
The total system of equation reads:

$$-4u_{\frac{1}{2}} + \frac{4}{3}u_{\frac{3}{2}} = \frac{-K\Delta x^2}{A} - \frac{8}{3}\bar{u}_0, \quad (54)$$

$$u(x_{i+\frac{3}{2}}) + u(x_{i-\frac{1}{2}}) - 2u(x_{i+\frac{1}{2}}) = \frac{-K\Delta x^2}{A}, \quad i = 1, n-2 \quad (55)$$

$$-(2 + \frac{N_2}{N_3})u_{n-\frac{1}{2}} + (1 - \frac{N_2}{N_3})u_{n-\frac{3}{2}} = \frac{-K\Delta x^2}{A} - \frac{1}{N_3}\bar{u}_L. \quad (56)$$

For different values of α , the system has been solved and the solutions are depicted in figure 14:



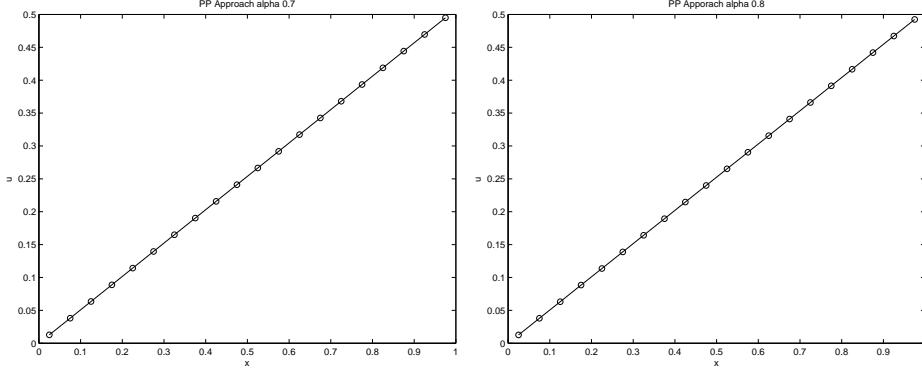


Figure 14: Solution With Cut Cell for different values of α

1.6 First Attempt: Finite Volume Method (Prof. Colella Approach)

1.6.1 Theory

The same problem stated in (1) is going to be solved here. We obtain an equation at each centroid of a cell.

$$\frac{\partial^2 u}{\partial x^2}|_{x_{i+\frac{1}{2}}} = \frac{-K}{A}, \quad (57)$$

$$\Leftrightarrow \frac{1}{\Delta x} \left(\frac{\partial u}{\partial x}|_R - \frac{\partial u}{\partial x}|_L \right) = \frac{-K}{A}, \quad (58)$$

where R is the next vertex to the right and L to the next vertex to the left.

At an inner element, the approximation of the second derivative reads:

$$\Leftrightarrow \frac{1}{\Delta x} \left(\frac{\partial u}{\partial x}|_R - \frac{\partial u}{\partial x}|_L \right) = \frac{1}{\Delta x} \left(\frac{1}{\Delta x} (u_{i+2} - u_i) - \frac{1}{\Delta x} (u_{i+1} - u_{i-1}) \right), \quad (59)$$

$$= \frac{1}{\Delta x^2} (u_{i+2} - u_i - u_{i+1} + u_{i-1}), \quad (60)$$

where u_{i-1} , u_i , u_{i+1} and u_{i+2} are the nodal (vertices) displacements.²

1.6.2 Results

The system has been solved with $A = 3$, $K = 0$, $L = 1$, $\bar{u}_0 = 0$ and $\bar{u}_L = 0.5$.³

²This was a major misunderstanding in the theory of finite volumes. The equations are written at the centroids, as well as the unknowns are held at the centroids and not at the node as done above. The wrong way to do it leads to an issue if the last element is a full. In this case we have more equations than unknowns, assuming that boundary conditions are prescribed on both ends. If one of the unknowns is a free end and we have a cut cell, we have less equations than unknowns. This is very obvious sign that there is a misunderstanding. The theory is not going to be derived based on this wrong assumption. The correct theory is going to be derived in section 1.7

³There is a mistake in the understanding of the theory. That is the explanation for the oscillatory behavior of the solution, that is not compatible with the properties of the solution to an elliptic equation. A solution to an elliptic equation is necessarily not oscillatory.

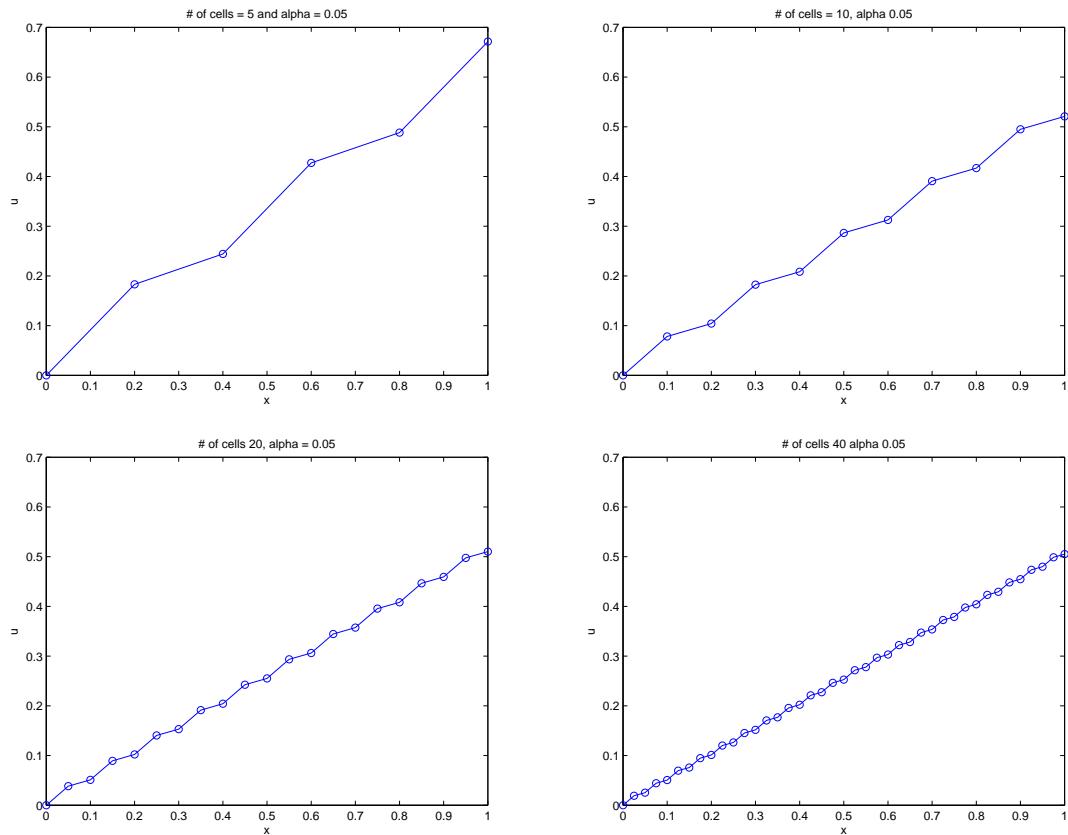
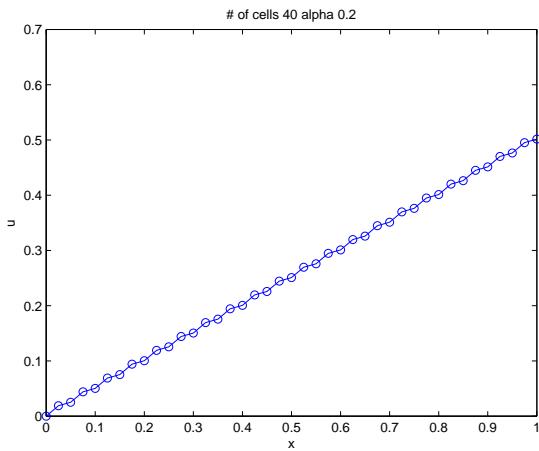


Figure 15: Convergence Study for $\alpha = 0.05$



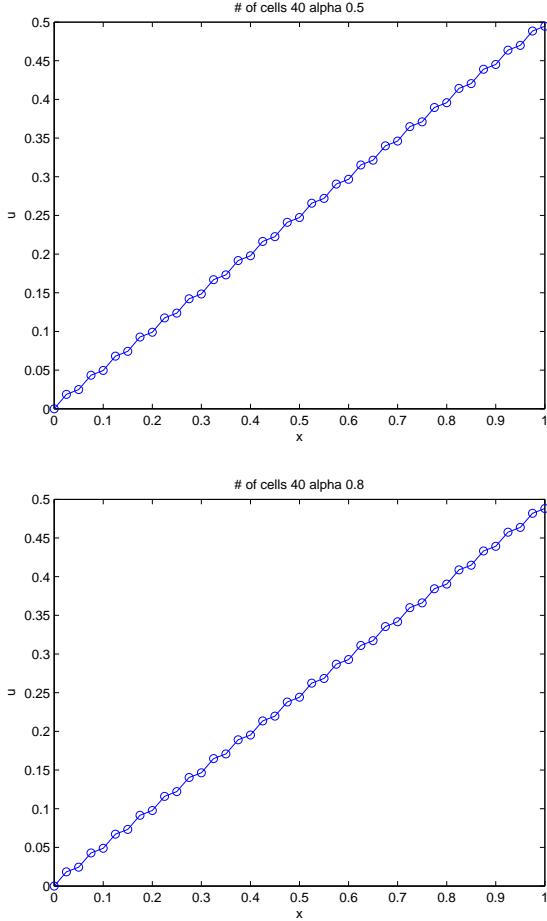


Figure 16: Solution With Cut Cell for different values of α

1.7 Second Attempt: Finite Volume Method (Prof. Colella Approach)

1.7.1 Theory

The same problem stated in (1) is going to be solved here. We obtain an equation at each centroid of a cell.

$$\frac{\partial^2 u}{\partial x^2} \Big|_{x_{i+\frac{1}{2}}} = \frac{-K}{A}, \quad (61)$$

$$\Leftrightarrow \frac{1}{\Delta x} \left(\frac{\partial u}{\partial x} \Big|_R - \frac{\partial u}{\partial x} \Big|_L \right) = \frac{-K}{A}, \quad (62)$$

$$\Leftrightarrow \frac{\partial u}{\partial x} \Big|_R - \frac{\partial u}{\partial x} \Big|_L = \frac{-K \Delta x}{A}, \quad (63)$$

The first derivative at the vertices are defined as follows:

$$\frac{\partial u}{\partial x} = \frac{1}{\Delta x} (u_{i+1} - u_i), \quad (64)$$

where u_{i+1} and u_i are the displacement on the neighbor centroids to the cell (see figure 17).

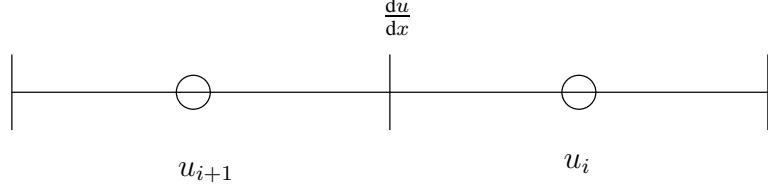


Figure 17: Dependency of the Derivative at the Vertex on the Centroids Values

We have an equation per control volume. At a non-boundary element, the expression for the second derivative reads:

$$\frac{1}{\Delta x}(u_{i+1} - u_i) - \frac{1}{\Delta x}(u_i - u_{i-1}) = \frac{-K\Delta x}{A}, \quad (65)$$

$$\Leftrightarrow u_{i+1} - 2u_i + u_{i-1} = \frac{-K\Delta x^2}{A}, \quad (66)$$

(67)

where i is the index of the centroid of the current control volume.⁴ the equation at the boundary elements can not be written as straightforward. We still have:

$$\frac{\partial u}{\partial x}|_R - \frac{\partial u}{\partial x}|_L = \frac{-K\Delta x}{A}. \quad (68)$$

If we consider the left boundary of the one-dimensional domain, the term $\frac{\partial u}{\partial x}|_R$ is defined as before in equation (64). The first derivative at the left boundary, i.e. $\frac{\partial u}{\partial x}|_L$, which corresponds to the end of the physical and finite volume domain, has to be determined from an interpolation over the domain end and the two first cells' centroids. the interpolation reads

⁴Note here, that this equation is exactly the same as the one obtained in section 1.5

as follows:

$$\frac{\partial u}{\partial x}|_L = N_{1,x}(0)u_e + N_{2,x}(0)u_1 + N_{3,x}(0)u_2, \quad (69)$$

$$N_1 = \frac{(x - \frac{\Delta x}{2})(x - \frac{3\Delta x}{2})}{\frac{3\Delta x^2}{4}}, \quad (70)$$

$$N_2 = -\frac{x(x - \frac{3\Delta x}{2})}{\frac{1}{2}\Delta x^2}, \quad (71)$$

$$N_3 = \frac{x(x - \frac{\Delta x}{2})}{\frac{3}{2}\Delta x^2}, \quad (72)$$

$$N_{1,x}(x) = \frac{2x - 2\Delta x}{\frac{3}{4}\Delta x^2}, \quad (73)$$

$$N_{2,x}(x) = \frac{-2x + \frac{3\Delta x}{2}}{\frac{\Delta x^2}{2}}, \quad (74)$$

$$N_{3,x}(x) = \frac{2x - \frac{\Delta x}{2}}{\frac{3}{2}\Delta x^2}, \quad (75)$$

where u_e is the displacement prescribed at the left boundary, u_1 and u_2 are the displacements of the cells' centroids.

The equation at the last cell reads:

$$\frac{\partial u}{\partial x}|_R - \frac{\partial u}{\partial x}|_L = \frac{-K\alpha\Delta x}{A}, \quad (76)$$

where α is the length of the physical domain compared to the cell size. The derivative to the left of the centroid is given directly by (64). The derivative to the right of the centroid might fall on a vertex or at the end of the cut cell. The derivative at the end of the physical domain,

whether it falls on a vertex or not, is obtained by:

$$\frac{\partial u}{\partial x}|_R = N_{1,x} \left(\frac{3}{2} \Delta x \right) u_{n-2} + N_{2,x} \left(\frac{3}{2} \Delta x \right) u_{n-1} + N_{3,x} \left(\frac{3}{2} \Delta x \right) \bar{u}, \quad (77)$$

$$N_1 = \frac{(x - \Delta x)(x - (\frac{3}{2} + \alpha)\Delta x)}{(\frac{3}{2} + \alpha)\Delta x^2}, \quad (78)$$

$$N_2 = -\frac{x(x - (\frac{3}{2} + \alpha)\Delta x)}{(\frac{1}{2} + \alpha)\Delta x^2}, \quad (79)$$

$$N_3 = \frac{x(x - \Delta x)}{(\frac{3}{2} + \alpha)(\frac{1}{2} + \alpha)\Delta x^2}, \quad (80)$$

$$N_{1,x}(x) = \frac{2x - (\frac{5}{2} + \alpha)\Delta x}{(\frac{3}{2} + \alpha)\Delta x^2}, \quad (81)$$

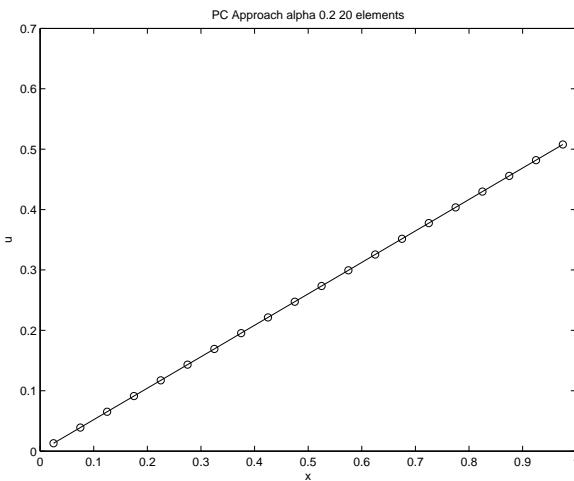
$$N_{2,x}(x) = -\frac{2x - (\frac{3}{2} + \alpha)\Delta x}{(\frac{3}{2} + \alpha)\Delta x}, \quad (82)$$

$$N_{3,x}(x) = \frac{2x - \Delta x}{(\frac{3}{2} + \alpha)(\frac{1}{2} + \alpha)\Delta x^2}, \quad (83)$$

where u_e is the displacement prescribed at the end of the physical domain, u_{n-1} and u_{n-2} are the centroids of the neighbor full cells to the last cell.

1.7.2 Results

For different values of α , the solution is given in figure 18.



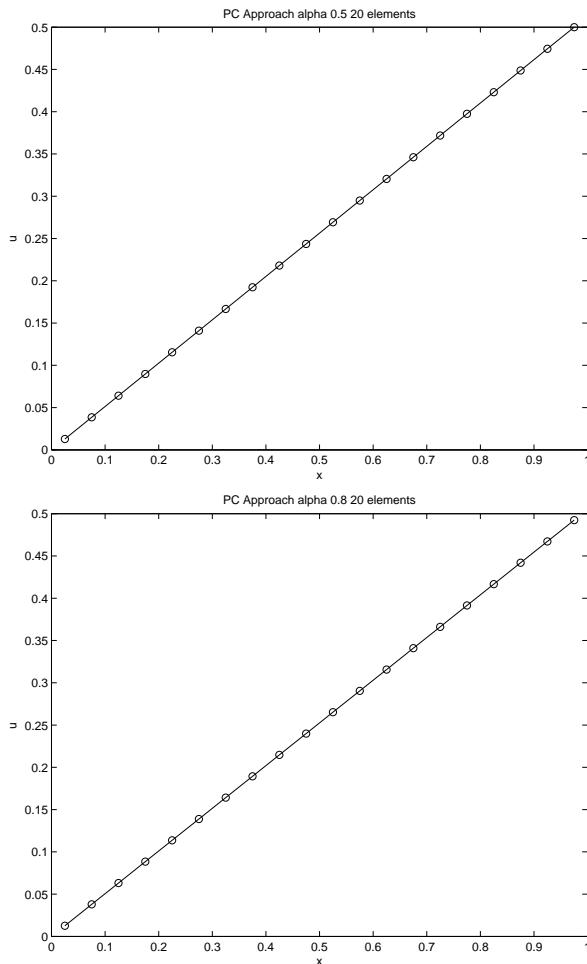
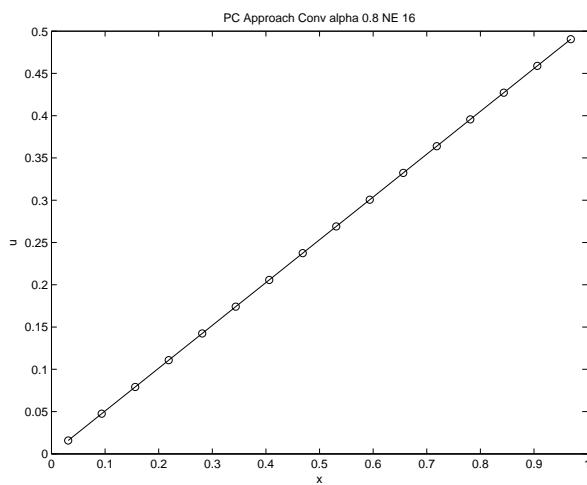


Figure 18: Solution for different values of α

While fixing the value of α , a convergence study with increasing the total number of elements is conducted. The results are depicted in figure 19.



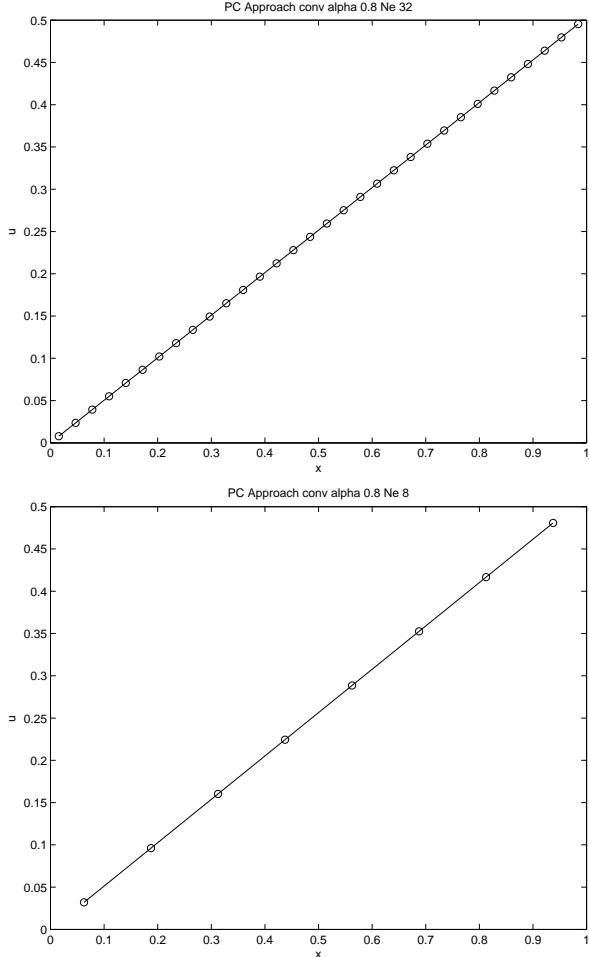


Figure 19: Convergence Study for $\alpha = 0.8$

1.8 Consideration of the PDE to the last element (Finite Element Method)

1.8.1 Results

Problem (1) is going to be solved. The following parameters are used:

$$L = 1, \quad h = 0.25, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (84)$$

$$\bar{u}|_L = 0.5, \quad K = 0, \quad \alpha = 0.9, \quad (85)$$

where h is the element size.

The PDE contribution is considered to the entire element, even though part of it is not included in the real physical domain. The assembling of the last element contribution to the global system helps propagating the information about the presence of a boundary condition. The boundary condition equations replace the PDE equations to the outer left and right nodes, noting that these nodes are not of interest for us because they do not really belong

to the original domain. These nodes have been created to have a fair match with the finite volume method solution positions. The PDE contribution is also considered for the entire first element, even though the left part passes the actual boundary of the domain. The solution is depicted in figure 20.

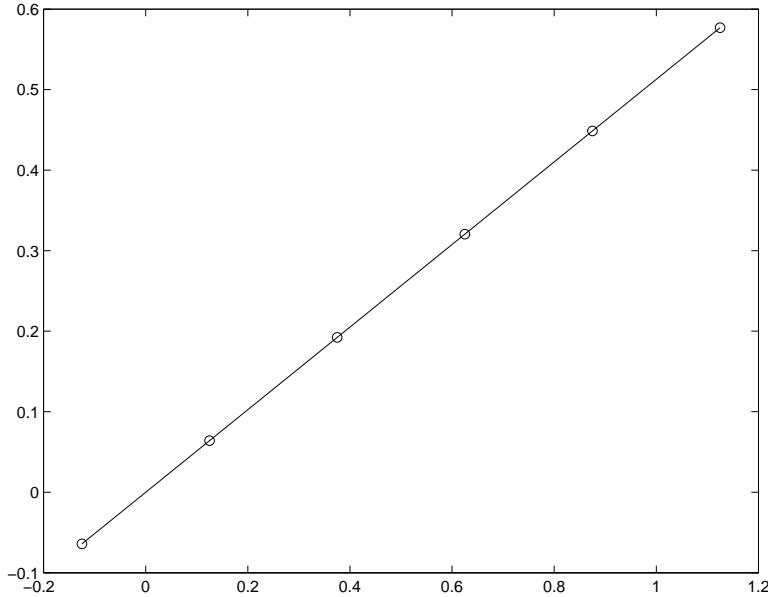


Figure 20: Finite Element with Consideration of PDE

2 Comparison Of All Of The Methods

Problem (1) is going to be solved using all of the methods discussed above. The following parameters are used:

$$L = 1, \quad , ne = 4, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (86)$$

$$\bar{u}|_L = 0.5, \quad K = 0, \quad \alpha = 0.9. \quad (87)$$

We are interested in comparing the outcomeing system of equations that has to be solved and hopefully recognize.⁵

2.1 PP Finite Volume

$S =$

$$\begin{array}{cccc} -4.0000 & 1.3333 & 0 & 0 \\ 1.0000 & -2.0000 & 1.0000 & 0 \end{array}$$

⁵In the following, s is the stiffness matrix, f is the force vector and u_n is the solution.

```
0      1.0000    -2.0000     1.0000
0          0     1.4286    -5.0000
```

f =

```
0
0
0
-1.7857
```

The unknowns are at the centroids of the volumes:

u_n =

```
0.0641
0.1923
0.3205
0.4487
```

2.2 PC Finite Volume

s =

```
-16.0000    5.3333      0      0
 1.0000   -2.0000     1.0000      0
  0      1.0000   -2.0000     1.0000
  0      2.3333   -2.8571    -4.0000
```

f =

```
0
0
0
-2.2619
```

The unknowns are at the centroids of the volumes:

u_n =

```
0.0641
0.1923
0.3205
0.4487
```

2.3 Finite Element 2 Elements Coupling

Before application of the left boundary conditions:

s =

```
12.0000 -12.0000      0      0      0
-12.0000  24.0000 -12.0000      0      0
      0 -12.0000   24.0000 -12.0000      0
      0      0 -12.0000   12.0000      0
      0      0 -0.0526    0.2222   1.0000
```

f =

```
0
0
0
0
0.5848
```

After application of the left boundary conditions:

s =

```
24.0000 -12.0000      0      0
-12.0000  24.0000 -12.0000      0
      0 -12.0000   12.0000      0
      0 -0.0526    0.2222   1.0000
```

f =

```
0
0
0
0.5848
```

The unknowns are at the nodes.

u_n =

```
0
0
0
0.5848
```

2.4 Finite Element 3 Elements Coupling

Before application of the left boundary conditions:

s =

```
12.0000 -12.0000      0      0      0
-12.0000  24.0000 -12.0000      0      0
      0 -12.0000   24.0000 -12.0000      0
      0       0 -12.0000   12.0000      0
      0     0.0285 -0.1755   0.3705  1.1115
```

f =

```
0
0
0
0
0.5000
```

After application of the left boundary conditions:

s =

```
24.0000 -12.0000      0      0
-12.0000  24.0000 -12.0000      0
      0 -12.0000   12.0000      0
    0.0285 -0.1755   0.3705  1.1115
```

f =

```
0
0
0
0.5000
```

The unknowns are at the nodes.

u_n =

```
0
0
0
0.4498
```

2.5 Finite Element Tranair Approach With Consideration of the DE Contribution

2.5.1 Penalty Method

The solution is calculated with a penalty parameter of 1000.

Before application of the left boundary conditions:

s =

```
-12.0000  12.0000      0      0      0
 12.0000 -24.0000  12.0000      0      0
   0    12.0000 -24.0000  12.0000      0
   0        0    12.0000 169.9759 257.0866
   0        0          0 257.0866 303.8509
```

f =

```
0
0
0
219.5312
280.4688
```

After application of the left boundary conditions:

s =

```
-24.0000  12.0000      0      0
 12.0000 -24.0000  12.0000      0
   0    12.0000 169.9759 257.0866
   0        0 257.0866 303.8509
```

f =

```
0
0
219.5312
280.4688
```

The unknowns are at the nodes.

```
u_n =
  0
 0.1498
```

0.2996
0.4494
0.5428

It is to be noted here that the finite element method solution is not accurate. But also, we could not compare correctly because of the misplacement of the unknowns in the methods, i.e. the finite element method positions the unknowns at cell edges and the finite volume method positions them at the cell centers. This has to be taken into account in future investigations.

2.5.2 Lagrange Multiplier

The Lagrange multiplier approach has not been implemented, because it is just going to be a different method for imposing the Dirichlet boundary conditions, which we know that it is going to work. We do not see any urgent purpose for implementing this because it would not lead to new conclusions regarding our study.

2.6 Remarks

It is to be noted here that the finite element method solution is not accurate. But also, we could not compare correctly because of the misplacement of the unknowns in the methods, i.e. the finite element method positions the unknowns at cell edges and the finite volume method positions them at the cell centers. This has to be taken into account in future investigations. Therefore, one can force the position of the unknowns of the finite element method at the cell centers. For this purpose, one expands the domain to the right and to the left to maintain the same mesh size for all methods (see figure 21).

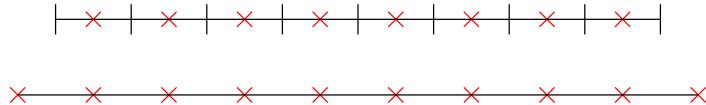


Figure 21: Position of the unknowns: Finite Volume Method (upper), Finite Element Method (bottom)

3 Comparisons: New Attempt

Problem (1) is going to be solved using all of the methods discussed above. The following parameters are used:

$$L = 1, \quad h = 0.25, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (88)$$

$$\bar{u}|_L = 0.5, \quad K = 0, \quad \alpha = 0.9, \quad (89)$$

where h is the element size.

3.1 PP Finite Volume

s =

```
-4.0000    1.3333      0      0
 1.0000   -2.0000     1.0000      0
  0       1.0000   -2.0000     1.0000
  0           0     1.4286   -5.0000
```

f =

```
 0
 0
 0
-1.7857
```

The unknowns are at the centroids of the volumes:

u_n =

```
 0.0641
 0.1923
 0.3205
 0.4487
```

3.2 Scaled PP Finite Volume

After obtaining the finite element solution, we want to fairly compare with the vector and matrix entries of the finite volume method to hopefully extract an analogy. The finite volume system is scaled by 12.

s =

```
-48.0000    15.9996      0      0
 12.0000   -24.0000     12.0000      0
  0       12.0000   -24.0000     12.0000
  0           0     17.1432   -60.0000
```

f =

```
 0
 0
 0
-21.4284
```

The unknowns are at the centroids of the volumes:

$u_n =$

0.0641
0.1923
0.3205
0.4487

3.3 PC Finite Volume

$s =$

-16.0000	5.3333	0	0
1.0000	-2.0000	1.0000	0
0	1.0000	-2.0000	1.0000
0	2.3333	-2.8571	-4.0000

$f =$

0
0
0
-2.2619

The unknowns are at the centroids of the volumes:

$u_n =$

0.0641
0.1923
0.3205
0.4487

3.4 Scaled PC Finite Volume

After obtaining the finite element solution, we want to fairly compare the vector and matrix entries to hopefully extract an analogy. The finite volume system is scaled by 12.

$s =$

-192.0000	63.9996	0	0
12.0000	-24.0000	12.0000	0
0	12.0000	-24.0000	12.0000
0	27.9996	-34.2852	-48.0000

f =

0
0
0
-27.1428

The unknowns are at the centroids of the volumes:

u_n =

0.0641
0.1923
0.3205
0.4487

3.5 Finite Element 2 Elements Coupling With PDE contribution

s =

-3.0000 -6.0000 1.0000 0 0 0
-12.0000 24.0000 -12.0000 0 0 0
0 -12.0000 24.0000 -12.0000 0 0
0 0 -12.0000 24.0000 -12.0000 0
0 0 0 -12.0000 24.0000 -12.0000
0 0 0 -0.4286 3.0000 1.0000

f =

0
0
0
0
0
1.7857

The unknowns are at the cell centers.

u_n =

-0.0641

0.0641
0.1923

```
0.3205
0.4487
-----
0.5769
```

3.6 Finite Element Tranair Approach

s =

```
-12.0000 12.0000 0 0 0 0
12.0000 -24.0000 12.0000 0 0 0
0 12.0000 -24.0000 12.0000 0 0
0 0 12.0000 -24.0000 12.0000 0
0 0 0 12.0000 236.3819 252.3994
0 0 0 0 252.3994 246.8194
```

f =

```
0
0
0
0
250.3906
249.6094
```

The unknowns are at the cell centers.

u_n =

```
-0.0528
-----
0.0528
0.1584
0.2640
0.3697
-----
0.6333
```

The array entries presented above have been consistently implemented such that negative definiteness is given. For the sake of convenience, the signs have been changed such that positive definiteness is obtained.

4 Comparison Study

After looking at the results obtained in section 3, one is intrigued to know about the extend of the similarities between the methods used, i.e. one wants to compare as many digits as possible in the different solutions.

Since the finite element approach using a two-elements-coupling and considering the partial differential equation contribution to the entire physical domain led to an exactly same answer as the finite volume one, we are curious to know about the extend of correctness as well as the effect of the ratio α on the solution.

Now, the domain is extended to a half of the element size to the left to match the finite volume unknowns position. Therefore, there has been introduced a similar ratio to α that does always have the value of 0.5. It would also be good to know if the extension of the left boundary of the finite element domain does affect the solution. Problem (1) is going to be solved using all of the methods discussed above. The following parameters are used:

$$L = 1, \quad h = 0.25, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (90)$$

$$\bar{u}|_L = 0.5, \quad K = 0, \quad \alpha = 0.9, \quad (91)$$

where h is the element size.

4.1 PP Finite Volume

$S =$

$$\begin{array}{cccc} 4.0000 & -1.3333 & 0 & 0 \\ -1.0000 & 2.0000 & -1.0000 & 0 \\ 0 & -1.0000 & 2.0000 & -1.0000 \\ 0 & 0 & -1.4286 & 5.0000 \end{array}$$

$f =$

$$\begin{array}{c} 0 \\ 0 \\ 0 \\ 1.7857 \end{array}$$

The unknowns are at the centroids of the volumes:

$u_n =$

$$\begin{array}{c} 0.064102564102564 \\ 0.192307692307692 \\ 0.320512820512820 \\ 0.448717948717949 \end{array}$$

' The results are depicted in figure 22.

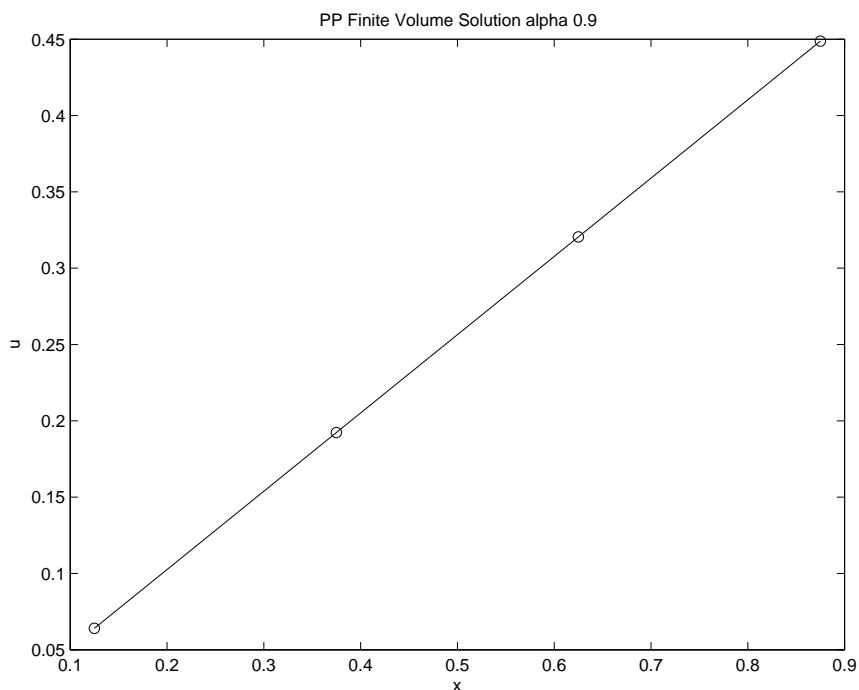


Figure 22: Solution with Finite Volume PP

4.2 PC Finite Volume

s =

$$\begin{matrix} 16.0000 & -5.3333 & 0 & 0 \\ -1.0000 & 2.0000 & -1.0000 & 0 \\ 0 & -1.0000 & 2.0000 & -1.0000 \\ 0 & -2.3333 & 2.8571 & 4.0000 \end{matrix}$$

f =

$$\begin{matrix} 0 \\ 0 \\ 0 \\ 2.2619 \end{matrix}$$

The unknowns are at the centroids of the volumes:

u_n =

$$\begin{matrix} 0.064102564102564 \\ 0.192307692307692 \end{matrix}$$

0.320512820512821
0.448717948717949

The results are depicted in figure 23.

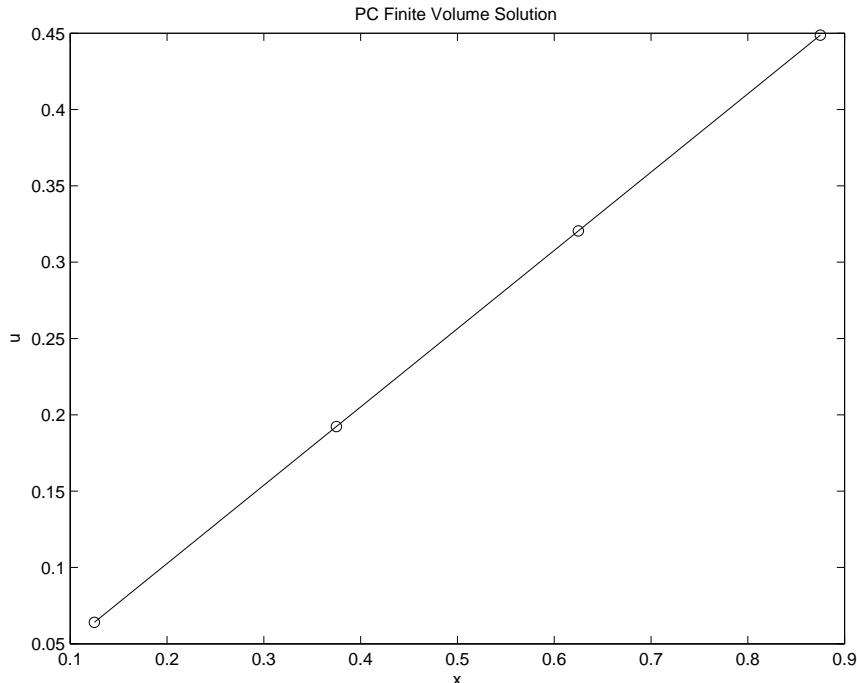


Figure 23: Solution with Finite Volume PC

4.3 Finite Element 2 Elements Coupling With Full PDE contribution

The PDE contribution is considered to the entire first and last element as if they would have belonged to the physical domain.

$S =$

$$\begin{matrix} 3.0000 & 6.0000 & -1.0000 & 0 & 0 & 0 \\ -12.0000 & 24.0000 & -12.0000 & 0 & 0 & 0 \\ 0 & -12.0000 & 24.0000 & -12.0000 & 0 & 0 \\ 0 & 0 & -12.0000 & 24.0000 & -12.0000 & 0 \\ 0 & 0 & 0 & -12.0000 & 24.0000 & -12.0000 \\ 0 & 0 & 0 & -0.4286 & 3.0000 & 1.0000 \end{matrix}$$

$f =$

$$\begin{matrix} 0 \\ 0 \end{matrix}$$

```
0
0
0
1.7857
```

The unknowns are at the cell centers.

```
u_n =
```

```
-0.064102564102564
-----
0.064102564102564
0.192307692307693
0.320512820512821
0.448717948717950
-----
0.576923076923078
```

The results are depicted in figure 24.

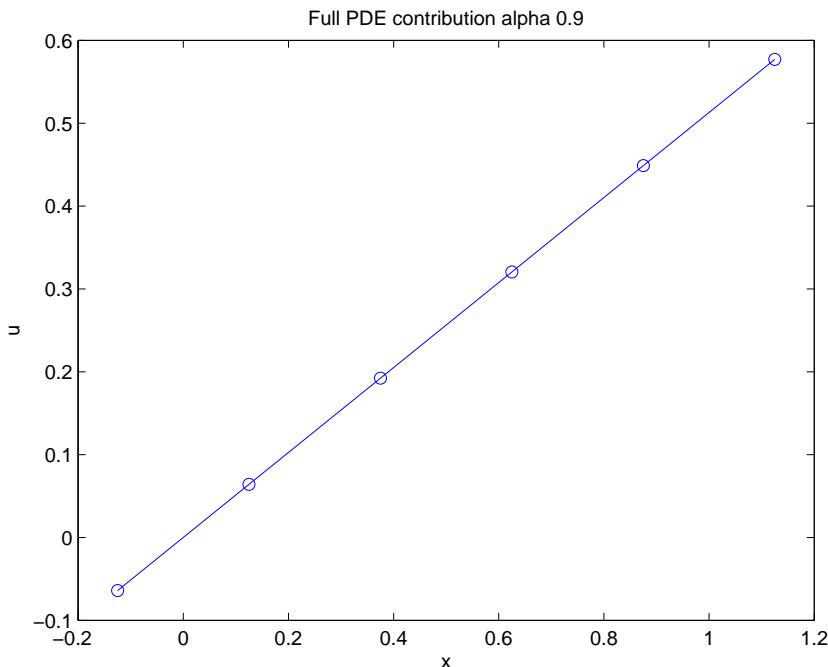


Figure 24: Solution with Full PDE Contribution

4.4 Finite Element 2 Elements Coupling With Scaled PDE contribution

The ratio of the physical domain length to the entire domain length is used to scale the PDE contribution to the first and last element as they do both expand the actual boundaries.

s =

3.0000	6.0000	-1.0000	0	0	0
-6.0000	18.0000	-12.0000	0	0	0
0	-12.0000	24.0000	-12.0000	0	0
0	0	-12.0000	24.0000	-12.0000	0
0	0	0	-12.0000	14.4000	-2.4000
0	0	0	-0.4286	3.0000	1.0000

f =

0
0
0
0
0
1.7857

The unknowns are at the cell centers.

u_n =

-0.104263206672845

0.081093605189991
0.173772011121409
0.266450417052827
0.359128822984245

0.822520852641336

The results are depicted in figure 25.

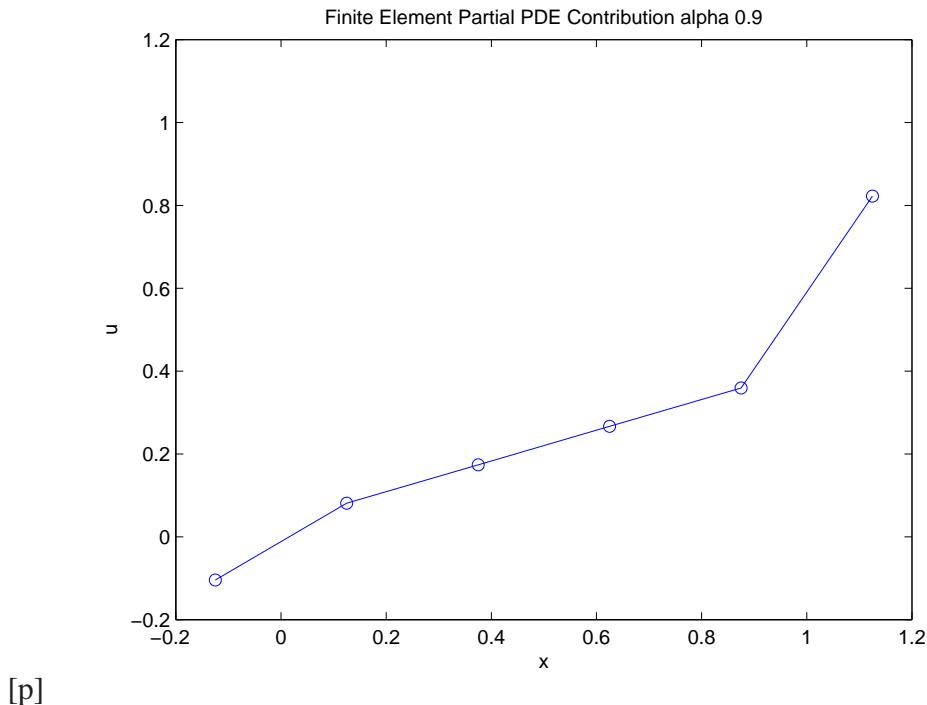


Figure 25: Solution with Partial PDE Contribution

4.5 Finite Element Tranair Approach

s =

3.0000	6.0000	-1.0000	0	0	0
-6.0000	18.0000	-12.0000	0	0	0
0	-12.0000	24.0000	-12.0000	0	0
0	0	-12.0000	24.0000	-12.0000	0
0	0	0	-12.0000	-236.3819	-252.3994
0	0	0	0	-252.3994	-246.8194

f =

0
0
0
0
-250.3906
-249.6094

The unknowns are at the cell centers.

```
u_n =
-----  

-0.110097737785369
-----  

0.085631573833065
0.183496229642282
0.281360885451498
0.379225541260715
-----  

0.623504896972005
```

The results are depicted in figure 26.

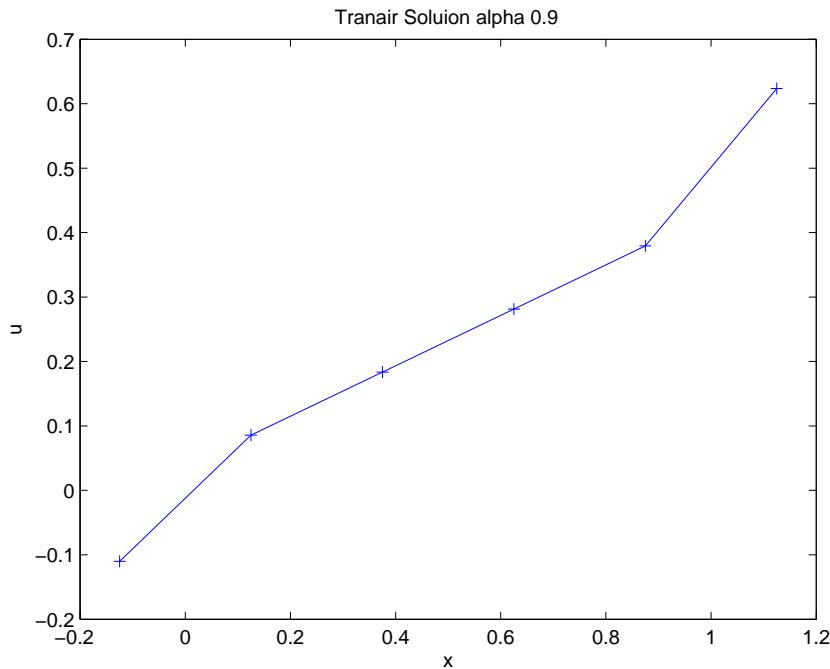


Figure 26: Solution with Tranair Approach

Positive definiteness of the system is convenient. We notice the negative entries on the last two rows of the stiffness matrix. Instead of adding the penalty terms, which collapse at the end to zero, the entries can be subtracted, i.e.

$$-\int A u_{,x} \xi_{,x} dx + \int K \xi dx - (\xi(u - \bar{u}))|_L = 0.$$

Therefore, we obtain positive diagonal entries, but a slight change in the solution, which will disappear when the element size decreases..

s =

3.0000	6.0000	-1.0000	0	0	0
-6.0000	18.0000	-12.0000	0	0	0
0	-12.0000	24.0000	-12.0000	0	0
0	0	-12.0000	24.0000	-12.0000	0
0	0	0	-12.0000	265.1819	247.5994
0	0	0	0	247.5994	251.6194

f =

0
0
0
0
250.3906
249.6094

The unknowns are at the cell centers.

u_n =

-0.109582971396014

0.085231199974678
0.182638285660024
0.280045371345370
0.377452457030717

0.620589674815624

4.6 Comments

It is interesting to see that the full consideration of the PDE to the last and first elements in the finite element approach leads to exactly the same answer as in the finite volume method. When multiplying the PDE contributions with a ratio α , the first and last finite elements are still considered entirely but with a partial stiffness. We therefore obtain a nonhomogeneous material distribution that leads to the nonlinearity we see in the solution both obtained by the Tranair approach and the finite element method with partial PDE contribution.

4.7 Finite Element 2 Elements Coupling With Full PDE contribution Linear BC Interpolation

At the beginning of this document, we have researched the use of different level of interpolation to enforce the boundary conditions. All levels have led to wrong results because we were considering the partially physical elements to be made of foam, i.e. they do not have any stiffness so no contribution to the PDE assembling. Now, we discovered that the full contribution of the PDE to the first and last elements do lead to a correct result, the same obtained from the finite volume method. In the one-dimensional case, the quadratic interpolation is computationally cheap, but extending this to multi-dimensions would be quite a bit of an effort. So, what if we replace the quadratic interpolation with a linear one and consistently consider the full contribution from the PDE?

s =

```
-0.5000 -0.5000 0 0 0 0
-12.0000 24.0000 -12.0000 0 0 0
0 -12.0000 24.0000 -12.0000 0 0
0 0 -12.0000 24.0000 -12.0000 0
0 0 0 -12.0000 24.0000 -12.0000
0 0 0 0 0.6000 0.4000
```

f =

```
0
0
0
0
0
0.5000
```

The unknowns are at the cell centers.

u_n =

```
-0.064102564102564
-----
0.064102564102564
0.192307692307692
0.320512820512820
0.448717948717949
-----
0.576923076923077
```

The results are depicted in figure 27.

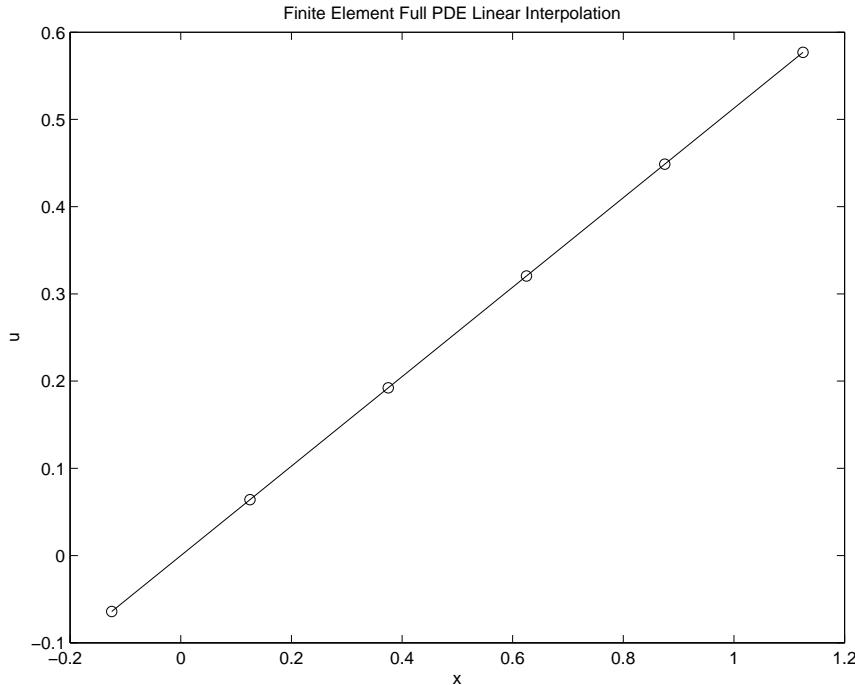


Figure 27: Solution with Full PDE Contribution Linear BC Interpolation

5 New Problem : Quadratic Solution

The solution we were looking at in the comparison studies was a purely linear solution due the lack of gravity, i.e. the right hand side K was zero. The linear interpolation for imposing the boundary conditions seemed to work, but we guess that this approach would fail if the solution itself would be quadratic. Therefore, we want to explore the choice of the level of interpolation depending on the solution type.

The problem we want to solve reads now:

$$L = 1, \quad h = 0.25, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (92)$$

$$\bar{u}|_L = 0.5, \quad K = -3, \quad \alpha = 0.9, \quad (93)$$

where h is the element size.

For reference, the finite volume solution is:

u_n =

```
0.010977564102564
0.079807692307692
0.211137820512821
0.404967948717949 ,
```

and depicted in figure 28.

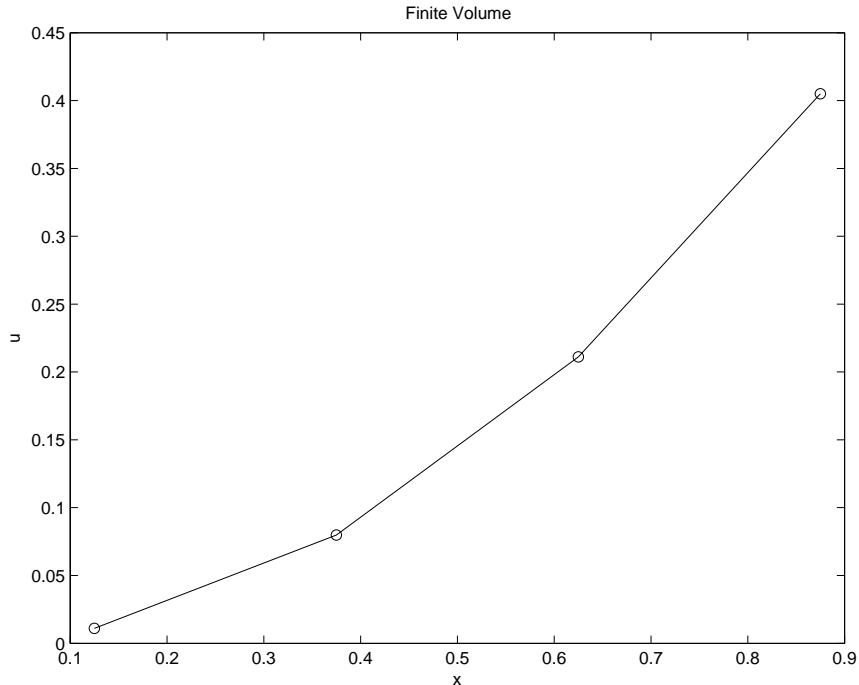


Figure 28: Quadratic Solution of Finite Volume Method

Imposing the boundary conditions means a linear interpolation leads to the following solution:

```
u_n =  
  
-0.003205128205128  
-----  
0.003205128205128  
0.072115384615385  
0.203525641025641  
0.397435897435897  
-----  
0.653846153846154 ,
```

that is depicted in figure 29.

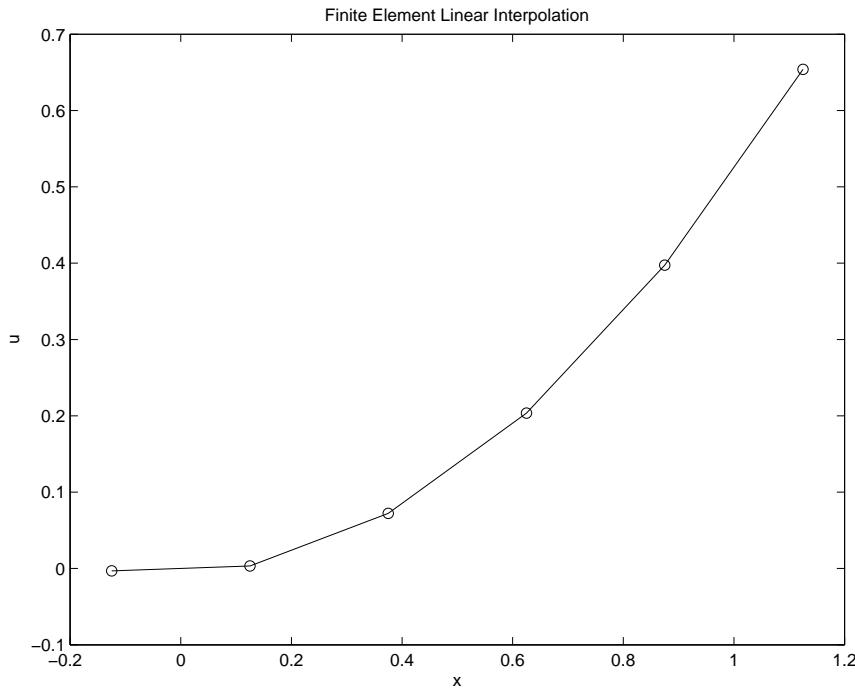


Figure 29: Quadratic Solution of Finite Element Method With Linear Interpolation of Boundary Conditions

The linear interpolation results are off in the order of 10^{-2} . This is sceptical because the error could be higher if the solution curvature would be higher.

Alternatively, using a quadratic interpolation for the boundary conditions, we obtain the following results:

`u_n =`

```
0.004647435897436
-----
0.010977564102564
0.079807692307693
0.211137820512821
0.404967948717950
-----
0.661298076923078
```

which are exactly the same as the one obtained from the finite volume method. The solution is depicted in figure 30.

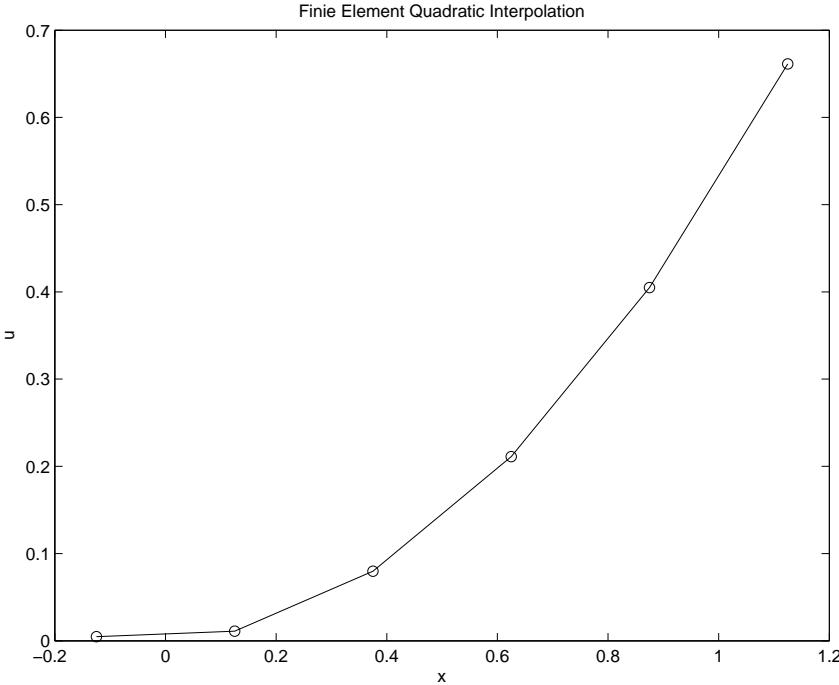


Figure 30: Quadratic Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions

6 Neumann Boundary Conditions

In the previous section, the idea of cut cells has been applied to boundaries with imposed Dirichlet boundary conditions. Now, we want to explore the treatment of Neumann boundary conditions within the finite element cut cell approach. The problem to be solved is slightly modified:

$$L = 1, \quad h = 0.05, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (94)$$

$$\frac{\partial u}{\partial x}|_L = 0.5, \quad K = -3, \quad \alpha = 0.9, \quad (95)$$

We have to maintain one Dirichlet boundary condition in order to obtain uniqueness of the solution.

6.1 Derivative of Quadratic Interpolation: Bug in analytical solution

Using the quadratic interpolation for the Dirichlet boundary condition on the cut cell edge, the Neumann boundary condition can be expressed as follows:

$$\frac{\partial u}{\partial x}|_L = \frac{\partial N_{n-2}}{\partial x}|_L u_{n-2} + \frac{\partial N_{n-1}}{\partial x}|_L u_{n-1} + \frac{\partial N_n}{\partial x}|_L u_n, \quad (96)$$

where

$$\frac{\partial N_{n-2}}{\partial x} = \frac{2x - 3\Delta x}{2\Delta x^2}, \quad (97)$$

$$\frac{\partial N_{n-1}}{\partial x} = -\frac{2x - 2\Delta x}{\Delta x^2}, \quad (98)$$

$$\frac{\partial N_n}{\partial x} = \frac{2x - \Delta x}{2\Delta x^2}. \quad (99)$$

For the specific problem we are looking at, the solution is depicted in figure 31.

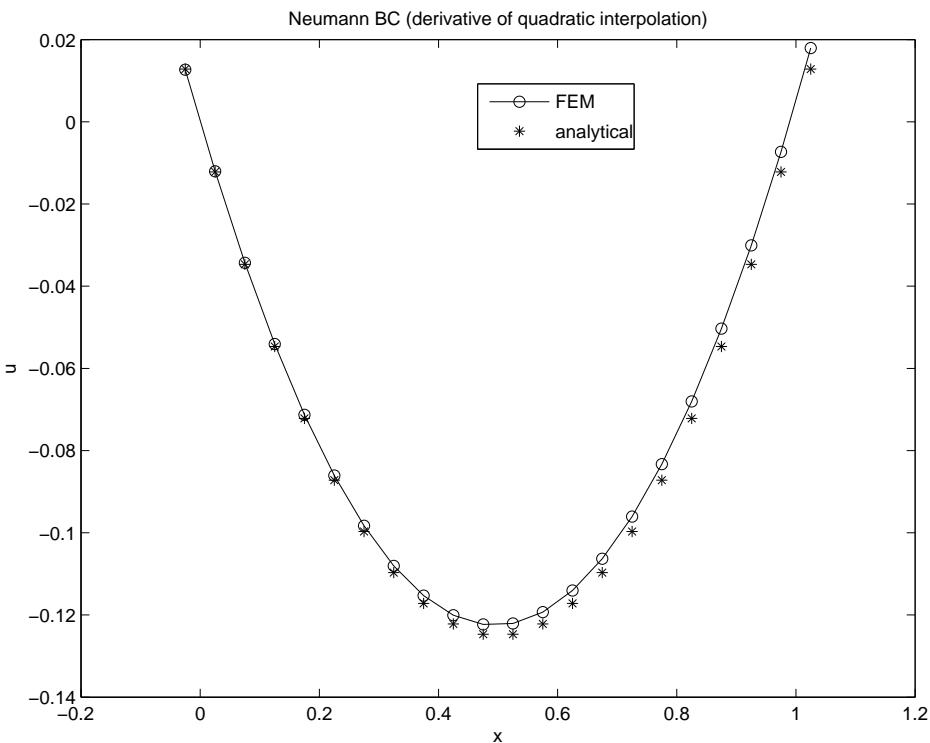


Figure 31: Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions).

We want to look carefully at the number of digits exactly satisfied by this comparison.⁶

⁶The finite element solutions do converge to the analytical solution that can be obtained easily to the problem we are solving. Convergence is obtained with decreasing the mesh size. For the specific coefficients we are choosing, and for the pure Dirichlet boundary conditions, the analytical solution reads $u = \frac{x^2}{2}$. For the Neumann boundary condition case, the analytical solution reads for the specific coefficients we chose above $u = \frac{x^2}{2} - \frac{1}{2}x$.

6.2 Derivative of Linear Interpolation: Bug in analytical solution

Using the linear interpolation for the Dirichlet boundary condition on the cut cell edge, the Neumann boundary condition can be expressed as follows:

$$\frac{\partial u}{\partial x}|_L = \frac{\partial N_{n-1}}{\partial x}u_{n-1} + \frac{\partial N_n}{\partial x}u_n, \quad (100)$$

where

$$\frac{\partial N_{n-1}}{\partial x} = \frac{-1}{\Delta x}, \quad (101)$$

$$\frac{\partial N_n}{\partial x} = \frac{1}{\Delta x}. \quad (102)$$

For the specific problem we are looking at, the solution is depicted in figure 32.

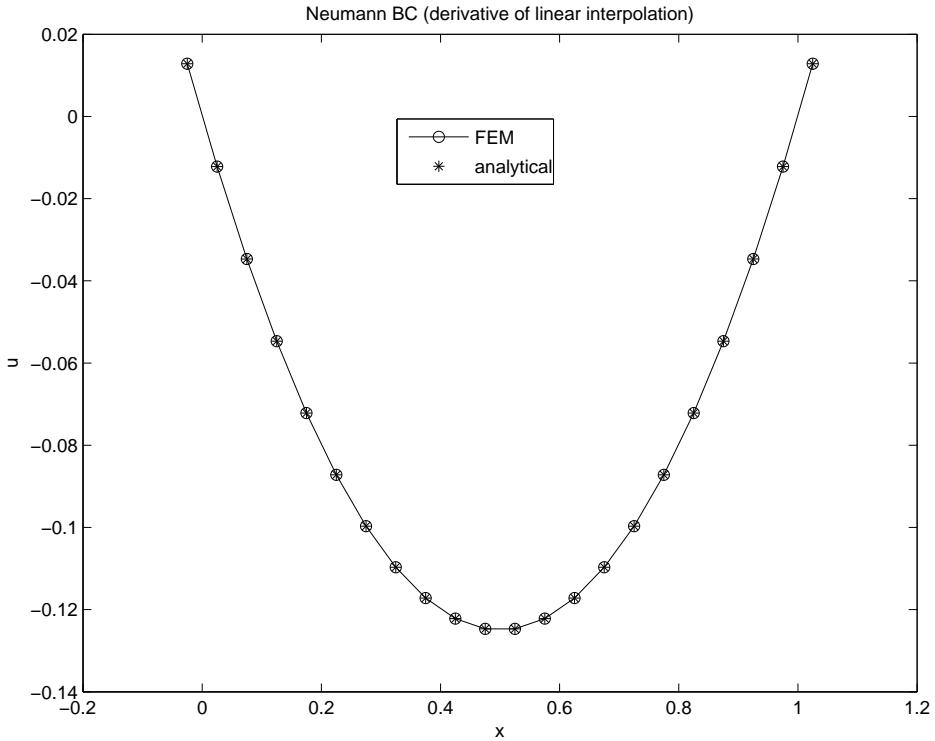


Figure 32: Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions).

Figure 32 shows that the linear interpolation (constant imposition of the Neumann boundary conditions) hits the analytical solution exactly. We want to look carefully at the number of digits exactly satisfied by this comparison. The analytical solution at the unknowns' positions reads:

u_nE =

```
0.012812500000000
-0.012187500000000
-0.034687500000000
-0.054687500000000
-0.072187500000000
-0.087187500000000
-0.099687500000000
-0.109687500000000
-0.117187500000000
-0.122187500000000
-0.124687500000000
-0.124687500000000
-0.122187500000000
-0.117187500000000
-0.109687500000000
-0.099687500000000
-0.087187500000000
-0.072187500000000
-0.054687500000000
-0.034687500000000
-0.012187500000000
0.012812500000000
```

The finite element solution at the unknowns' positions reads:

u_n =

```
0.012812500000000
-0.012187500000000
-0.034687500000000
-0.054687500000000
-0.072187500000000
-0.087187500000000
-0.099687500000000
-0.109687500000000
-0.117187500000000
-0.122187500000000
-0.124687500000000
-0.124687500000000
-0.122187500000000
-0.117187500000000
-0.109687500000000
-0.099687500000000
-0.087187500000000
-0.072187500000000
-0.054687500000000
-0.034687500000000
-0.012187500000000
-0.012187500000000
```

```
-0.109687500000001
-0.099687500000001
-0.087187500000001
-0.072187500000001
-0.054687500000001
-0.034687500000001
-0.012187500000001
 0.01281249999999
```

We do not notice any distinction between the two solutions to double precision accuracy.

6.3 Corrected: Derivative of Linear Interpolation and Quadratic Interpolation

Basically, the implementation of the linear and quadratic interpolations are bug-free. The Neumann boundary conditions have been imposed correctly in the finite element solution. However, the comparison with the analytical solution has not been done correctly. The analytical solution has accounted for a certain placement of the Neumann boundary condition, that is $\frac{\partial \bar{u}}{\partial x}|_L = 0.5$. This has been mistakenly given in the problem definition. Changing the value of α changes the end of the physical domain, and therefore the position of the Neumann boundary condition. We change the definition of the problem to the following:

$$L = 1, \quad , h = 0.05, \quad A = 3, \quad \bar{u}|_0 = 0.0, \quad (103)$$

$$\frac{\partial \bar{u}}{\partial x}|_{L_c} = 0.5, \quad K = -3, \quad \alpha = 0.9, L_c = L + h(\alpha - 1) = 0.995. \quad (104)$$

Thus, the analytical solution for this specific case reads:

$$u = \frac{x^2}{2} + 0.5 - L_c, \quad (105)$$

$$u = \frac{x^2}{2} - 0.495. \quad (106)$$

Once L_c changes, so do both the analytical solution and the finite element solution.

6.3.1 Derivative of Linear Interpolation

The finite element solution at the unknowns' positions reads:

`u_n =`

```
0.012812500000000
-0.012187500000000
-0.034687500000000
```

```
-0.054687500000000
-0.072187500000000
-0.087187500000000
-0.099687500000000
-0.109687500000000
-0.117187500000000
-0.122187500000000
-0.124687500000000
-0.124687500000000
-0.122187500000000
-0.117187500000000
-0.109687500000001
-0.099687500000001
-0.087187500000001
-0.072187500000001
-0.054687500000001
-0.034687500000001
-0.012187500000001
0.012812499999999
```

The analytical solution at the unknowns' positions reads:

$u_{nE} =$

```
0.012687500000000
-0.012062500000000
-0.034312500000000
-0.054062500000000
-0.071312500000000
-0.086062500000000
-0.098312500000000
-0.108062500000000
-0.115312500000000
-0.120062500000000
-0.122312500000000
-0.122062500000000
-0.119312500000000
-0.114062500000000
-0.106312500000000
-0.096062500000000
-0.083312500000000
-0.068062500000000
-0.050312500000000
```

```
-0.0300625000000000
-0.0073125000000000
 0.0179375000000000
```

We do not notice a deviation between the two solutions beginning from the third after comma digit (see also figure 33).

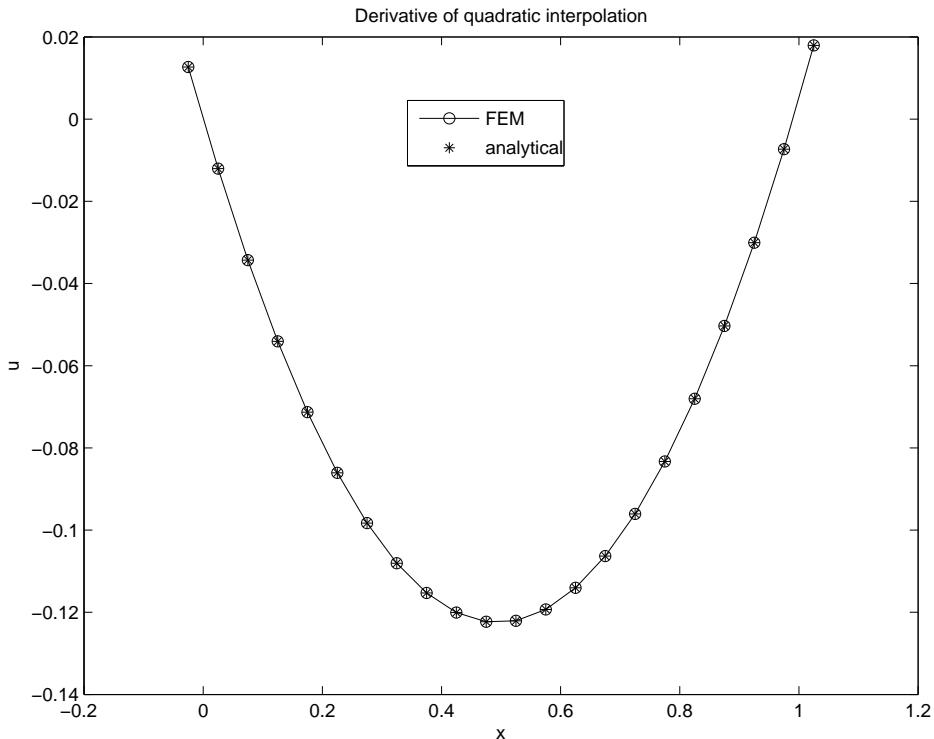


Figure 33: Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions).

6.3.2 Derivative of Quadratic Interpolation

The finite element solution at the unknowns' positions reads:

u_n =

```
0.0126875000000000
-0.0120625000000000
-0.0343125000000000
-0.0540625000000000
-0.0713125000000000
-0.0860625000000000
-0.0983125000000000
```

```
-0.108062500000001
-0.115312500000001
-0.120062500000001
-0.122312500000001
-0.122062500000001
-0.119312500000001
-0.114062500000001
-0.106312500000002
-0.096062500000002
-0.083312500000002
-0.068062500000002
-0.050312500000002
-0.030062500000002
-0.007312500000002
  0.01793749999998
```

The analytical solution at the unknowns' positions reads:

$u_{nE} =$

```
  0.012687500000000
-0.012062500000000
-0.034312500000000
-0.054062500000000
-0.071312500000000
-0.086062500000000
-0.098312500000000
-0.108062500000000
-0.115312500000000
-0.120062500000000
-0.122312500000000
-0.122062500000000
-0.119312500000000
-0.114062500000000
-0.106312500000000
-0.096062500000000
-0.083312500000000
-0.068062500000000
-0.050312500000000
-0.030062500000000
-0.007312500000000
  0.017937500000000
```

We do notice a deviation between the two solutions starting from the third after comma digit (see also figure 34).

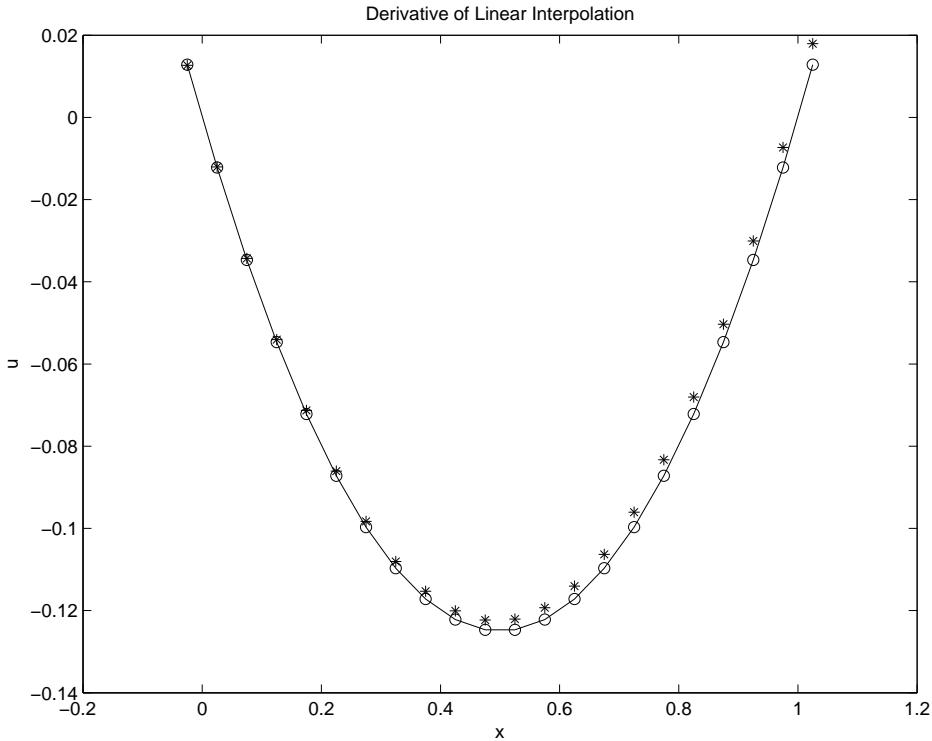


Figure 34: Neumann Boundary Conditions on End of Domain (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions).

6.3.3 Interpretations

The results obtained in the previous two sections is consistent for different values of α . The result, the derivative of a quadratic interpolation is more accurate than the derivative of a linear interpolation to impose the flux, we obtained does have an explanation. The analytical solution is quadratic. Therefore, the flux is a linear function. When we do impose a quadratic displacement interpolation, and thus a linear flux interpolation on the boundary, we do obtain quadratic convergence. In the case, where we use a linear interpolation of the displacement, and thus a constant flux function, we do not obtain convergence. The conclusion is that we have to remain consistent with the degree of the answer (linear, quadratic, cubic etc) when we impose the boundary conditions by interpolation in order to obtain convergence.

6.4 Finite Element Flux Solution

We know that the fluxes obtained from the finite element solution are discontinuous at the element boundaries.

The flux in an element is given by:

$$\frac{\partial u}{\partial x} = \sum_I \frac{\partial N_I}{\partial x} \frac{\partial \xi}{\partial x} u_I, \quad (107)$$

where the index I stands for the nodal index. The problem we have solved in the previous section results into a quadratic displacement solution, and therefore into a linear flux function. The flux solution are depicted in the figures for both Neumann boundary condition approaches.

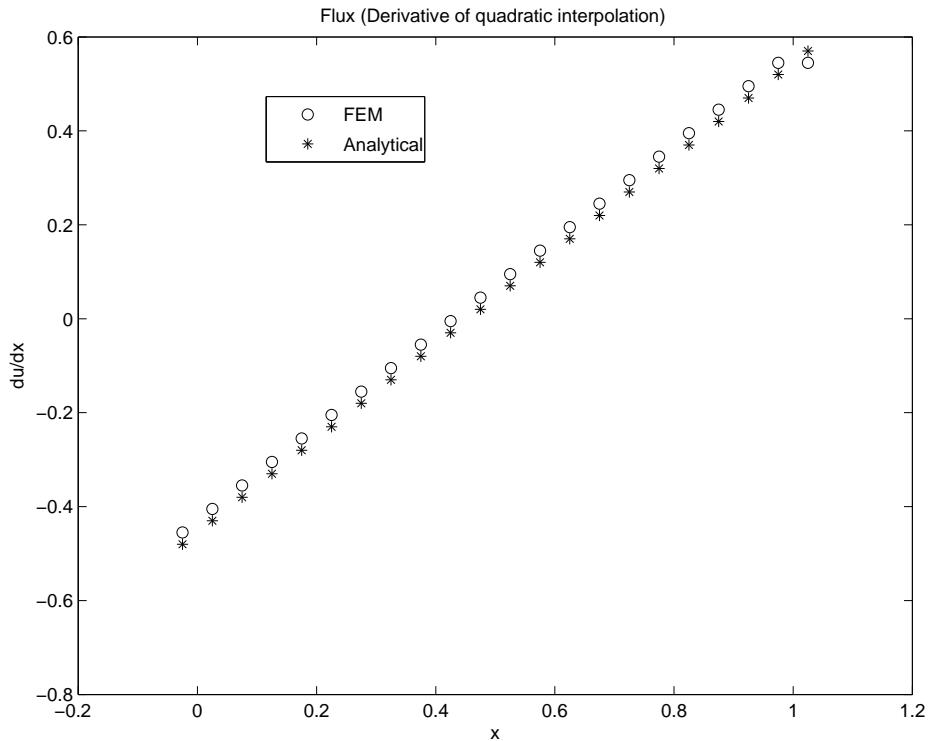


Figure 35: Flux Solution (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions).

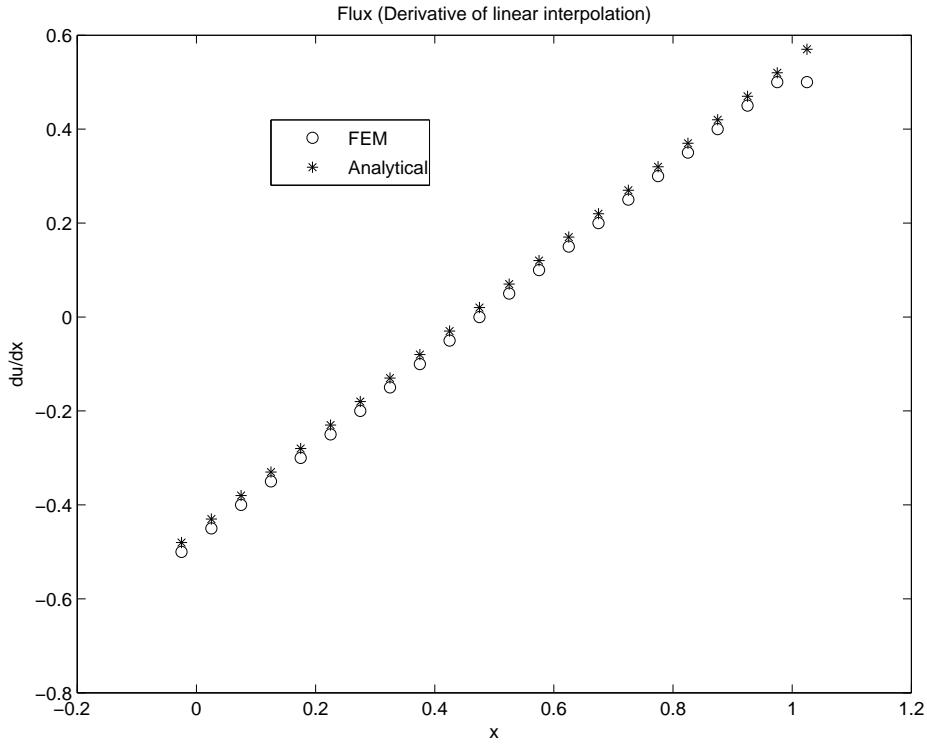


Figure 36: Flux Solution (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions).

We note here that both approaches are simply an approximation to the analytical solution. This behavior was not expected, since the quadratic interpolation of the Neumann boundary condition did deliver the exact solution. Therefore, one expects that its derivative would also be exact.

It is to remember that we are comparing the analytical solution with the finite element solution at sampling points, i.e. the nodes. We know that on many occasions the displacements, are most accurately sampled at the nodes and that the gradients are best sampled at some interior points [1]. The solution at the appropriate sampling points exhibit superconvergence according to the Herrmann theorem and show an error which decreases more rapidly than elsewhere.

While solving our problem, we do notice that nodes are the appropriate sampling points to obtain superconvergence. This happens to be the case because the weighting function used here contain the exact solution and because of the use of linear elements (solution at inner nodes of higher order elements tend to deviate from the exact solution and distortion is more likely to happen).

However, the nodes do not seem to be the best sampling points of the finite element flux solution and therefore should be sampled elsewhere in the interior of the elements. It can be shown that sampling somewhere near the center of each element is needed. The center of the element hits here the position of the gauss point needed to exactly integrate the analytical flux solution. We remember here that the flux solution is linear. Therefore, one gauss point

is enough to accurately integrate the flux solution over one element. At the gauss points, the flux is expected to be exact.

Using these information, we do compare the analytical and finite element fluxes at the element centers. The solutions are depicted in figures 37 and 38 .

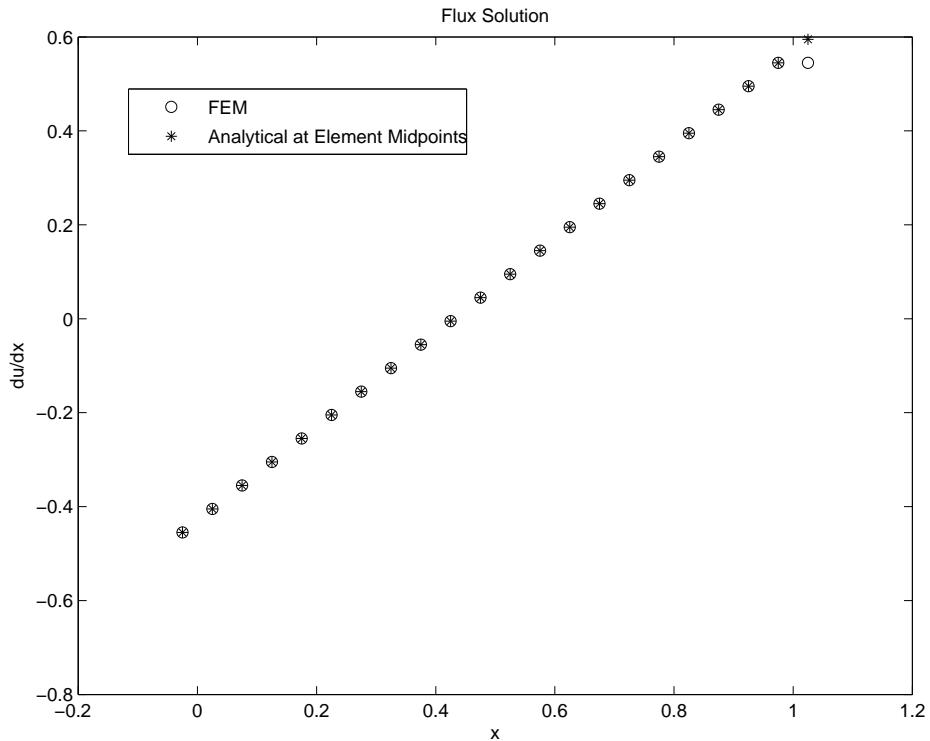


Figure 37: Flux Solution (Solution of Finite Element Method With Quadratic Interpolation of Boundary Conditions).

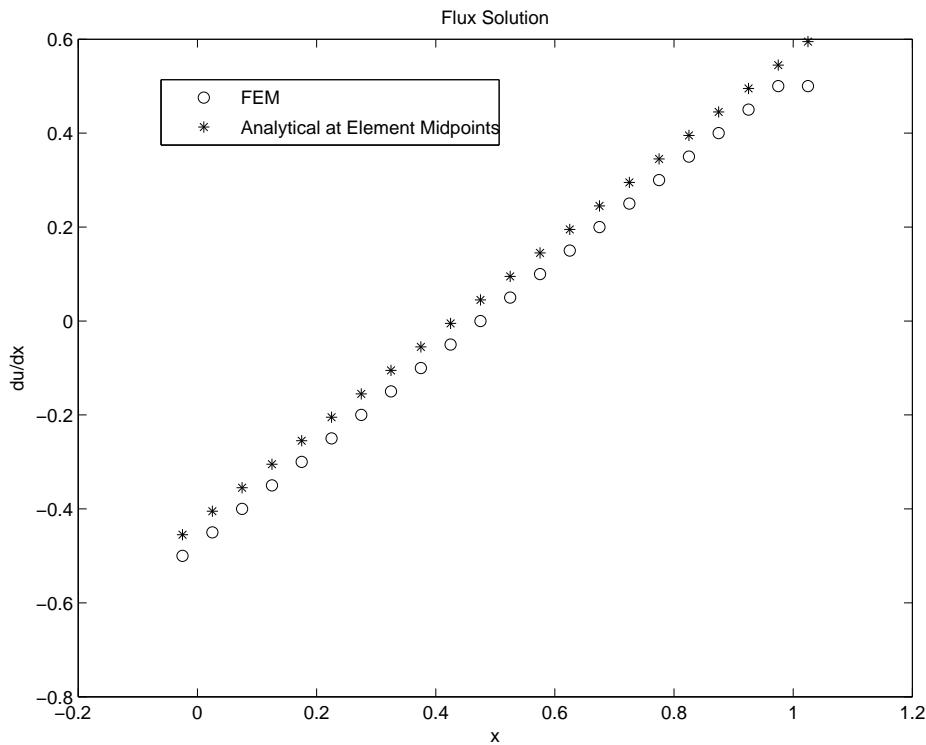


Figure 38: Flux Solution (Solution of Finite Element Method With Linear Interpolation of Boundary Conditions).

The analytical flux at the center points is given by:

```
du_nE =
-0.4550000000000000
-0.4050000000000000
-0.3550000000000000
-0.3050000000000000
-0.2550000000000000
-0.2050000000000000
-0.1550000000000000
-0.1050000000000000
-0.0550000000000000
-0.0050000000000000
0.0450000000000000
0.0950000000000000
0.1450000000000000
0.1950000000000000
0.2450000000000000
0.2950000000000000
```

```
0.3450000000000000
0.3950000000000000
0.4450000000000000
0.4950000000000000
0.5450000000000000
0.5950000000000000
```

The finite element flux at the center point for the quadratic interpolation approach reads:

```
du_n =
-0.455000000000001
-0.405000000000001
-0.355000000000001
-0.305000000000001
-0.255000000000001
-0.205000000000001
-0.155000000000001
-0.105000000000001
-0.055000000000001
-0.005000000000002
0.044999999999999
0.094999999999998
0.144999999999999
0.194999999999999
0.244999999999999
0.294999999999999
0.344999999999999
0.394999999999998
0.444999999999998
0.494999999999999
0.545000000000000
0.545000000000000
```

The finite element flux at the center point for the linear interpolation approach reads:

```
du_n =
-0.500000000000001
-0.450000000000001
-0.400000000000001
-0.350000000000001
-0.300000000000001
-0.250000000000001
```

```

-0.200000000000001
-0.150000000000001
-0.100000000000001
-0.050000000000001
-0.000000000000001
 0.049999999999999
 0.099999999999999
 0.149999999999999
 0.199999999999999
 0.249999999999999
 0.300000000000000
 0.349999999999999
 0.399999999999998
 0.449999999999999
 0.500000000000001
 0.500000000000001

```

The new sampling confirms that the quadratic interpolation of the solution, i.e. the linear interpolation of the flux leads to accurate results.

7 Two-Dimensional Approach

7.1 Canonical Problem: Circular Physical Domain

We want to solve the following problem using the finite element method on a rectangular two dimensional grid:

$$A \Delta u + f = 0, \quad (108)$$

$$A \nabla \cdot (\nabla u) + f = 0, \quad (109)$$

where A is a material constant and f a weighting constant function.

Note here that only the geometry is two-dimensional. The function we are solving for is scalar (not vectorial), i.e. the nodes do have one single degree of freedom.

Since the problem is two-dimensional, the function u does not change on the third dimension, and we can omit this dependence.

7.2 Analytical Solution

We can express our unknown u in terms of polar coordinates (r, θ) . Since the geometry is axisymmetric, the function u does not depend on the angular direction but on the radial

position.

$$A \nabla \cdot (\nabla u) + f = 0, \quad (110)$$

$$A \left(\underbrace{\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r}}_{=0} + \underbrace{\frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 f}{\partial z^2}}_{=0} \right) + f = 0, \quad (111)$$

$$A \left(\frac{d^2 u}{dr^2} + \frac{1}{r} \frac{du}{dr} \right) + f = 0, \quad (112)$$

$$\frac{d^2 u}{dr^2} + \frac{1}{r} \frac{du}{dr} = -\frac{f}{A}. \quad (113)$$

$$(114)$$

The homogeneous solution to the last equation is given by:

$$\text{Let } v = \frac{du}{dr}, \quad (115)$$

$$\text{The equation reads then: } \frac{dv}{dr} + \frac{1}{r}v = 0, \quad (116)$$

$$\frac{dv}{v} = -\frac{dr}{r}, \quad (117)$$

$$\int \frac{dv}{v} = \int -\frac{dr}{r} + \tilde{C}, \quad (118)$$

$$\ln(v) = -\ln(r) + \tilde{C}, \quad (119)$$

$$\ln(v) = \ln\left(\frac{1}{r}\right) + \tilde{C}, \quad (120)$$

$$e^{\ln(v)} = e^{\ln\left(\frac{1}{r}\right) + \tilde{C}}, \quad (121)$$

$$v = e^{\ln\left(\frac{1}{r}\right)} \underbrace{e^{\tilde{C}}}_{C_1}, \quad (122)$$

$$v = C_1 \frac{1}{r}, \quad (123)$$

$$\frac{du}{dr} = \frac{C_1}{r}, \quad (124)$$

$$\int du = C_1 \int \frac{dr}{r}, \quad (125)$$

$$u = C_1 \ln(r) + C_2. \quad (126)$$

$$(127)$$

Since a solution has to exist at the center of the circle, where $r = 0$, the term $\ln(r)$ blows up as it approaches the center. We omit this term by setting $C_1 = 0$. Thus, the homogeneous solution reads:

$$u_h = C_2. \quad (128)$$

We now have to solve for the particular solution from:

$$\frac{d^2u}{dr^2} + \frac{1}{r} \frac{du}{dr} = -\frac{f}{A}. \quad (129)$$

One assumes that the particular solution is of the form:

$$u = \frac{f}{A}(ar^2 + br), \quad (130)$$

where a and b are constants that have to be determined. Using this assumption, we obtain:

$$\frac{d^2u}{dr^2} + \frac{1}{r} \frac{du}{dr} = -\frac{f}{A}, \quad (131)$$

$$\frac{f}{A}(2a) + \frac{1}{r} \frac{f}{A}(2ar + b) = -\frac{f}{A}, \quad (132)$$

$$2a + \frac{1}{r}(2ar + b) = -1, \quad (133)$$

$$4a + \frac{1}{r}b = -1. \quad (134)$$

In order to obtain an acceptable solution at the center of the circle, we set $b = 0$. Finally, the particular solution reads:

$$4a = -1, \quad (135)$$

$$a = -\frac{1}{4}, \quad (136)$$

$$u_p = -\frac{f}{4A}r^2. \quad (137)$$

The total solution reads:

$$u = u_h + u_p = C_2 - \frac{f}{4A}r^2. \quad (138)$$

Given the Dirichlet boundary condition, where $u = \bar{u}$ at the boundary of the circular physical domain. Let's denote R to be the radius of the circle. Therefore, $u = \bar{u}$ for $r = R$.

$$\bar{u} = C_2 - \frac{f}{4A}R^2, \quad (139)$$

$$C_2 = \bar{u} + \frac{f}{4A}R^2, \quad (140)$$

$$u = \bar{u} + \frac{f}{4A}R^2 - \frac{f}{4A}r^2. \quad (141)$$

In the end, we realize that the solution is a combination of one-dimensional solutions since the solution varies only on each ray in the circle, and the solutions on the rays are independent of each other.

7.3 Weak Form

The weak form of the problem can be derived as follows:

$$\int_{\Omega} (A \nabla \cdot (\nabla u) + f)v \, d\Omega = 0, \quad (142)$$

$$\int_{\Omega} A \nabla \cdot (\nabla u)v + fv \, d\Omega = 0, \quad (143)$$

$$\int_{\Omega} A \nabla \cdot ((\nabla u)v) - A \nabla u \cdot \nabla v + fv \, d\Omega = 0, \quad (144)$$

$$\int_{\Omega} A \nabla \cdot ((\nabla u)v) \, d\Omega - A \int_{\Omega} \nabla u \cdot \nabla v + fv \, d\Omega = 0, \quad (145)$$

$$\int_{\partial\Omega} A((\nabla u)v)\mathbf{n} \, d(\partial\Omega) - A \int_{\Omega} \nabla u \cdot \nabla v + fv \, d\Omega = 0, \quad (146)$$

$$\int_{\partial\Omega} A((\nabla u)v)\mathbf{n} \, d(\partial\Omega) + fv \, d\Omega = A \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega. \quad (147)$$

$$(148)$$

The function u is imposed to pure Dirichlet conditions, s.t.:

$$u = \bar{u} \quad \text{on} \quad \partial\Omega_u, \quad (149)$$

$$(150)$$

Therefore, the weak form simplifies to:

$$\int_{\partial\Omega} fv \, d\Omega = A \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega. \quad (151)$$

7.4 Discretization

The following discretization convention has been used to create the vector-matrix entries:

$$\hat{\Phi}_1 = \frac{1}{4}(1 - \xi)(1 - \eta), \quad (152)$$

$$\hat{\Phi}_2 = \frac{1}{4}(1 + \xi)(1 - \eta), \quad (153)$$

$$\hat{\Phi}_3 = \frac{1}{4}(1 + \xi)(1 + \eta), \quad (154)$$

$$\hat{\Phi}_4 = \frac{1}{4}(1 - \xi)(1 + \eta), \quad (155)$$

$$\hat{\Phi}^e = \begin{pmatrix} \hat{\Phi}_1^e \\ \hat{\Phi}_2^e \\ \hat{\Phi}_3^e \\ \hat{\Phi}_4^e \end{pmatrix}, \quad (156)$$

$$D\hat{\Phi}^e = \begin{pmatrix} \frac{\partial \hat{\Phi}_1^e}{\partial \xi} & \frac{\partial \hat{\Phi}_1^e}{\partial \eta} \\ \frac{\partial \hat{\Phi}_2^e}{\partial \xi} & \frac{\partial \hat{\Phi}_2^e}{\partial \eta} \\ \frac{\partial \hat{\Phi}_3^e}{\partial \xi} & \frac{\partial \hat{\Phi}_3^e}{\partial \eta} \\ \frac{\partial \hat{\Phi}_4^e}{\partial \xi} & \frac{\partial \hat{\Phi}_4^e}{\partial \eta} \end{pmatrix}, \quad (157)$$

$$X^e = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{pmatrix}, \quad (158)$$

$$F = (X^e)D\hat{\Phi}^e, \quad (159)$$

$$C = F^T F, \quad (160)$$

$$J = \det(F). \quad (161)$$

Therefore, we obtain:

$$K_{ij}^e = \int_{\Omega_0^e} A D\hat{\Phi}^e C^{-1} (D\hat{\Phi}^e)^T J d\Omega, \quad (162)$$

$$R_i^e = \int_{\Omega_0^e} f \hat{\Phi}^e J d\Omega \quad (163)$$

7.5 Geometry

The geometry chosen for the canonical problem is a circle with defined center position and radius that is discretized using a rectangular grid (Figure 39).

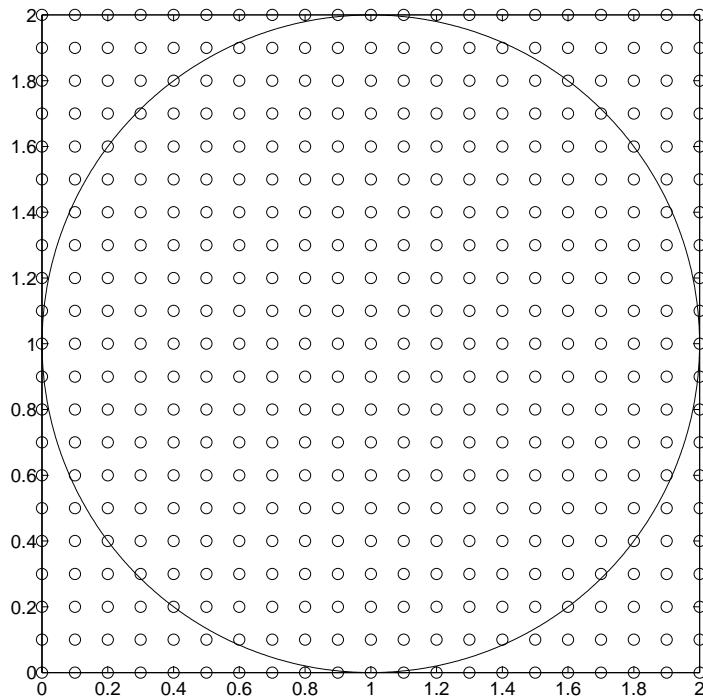


Figure 39: Circular Physical Domain and Cartersian Grid

7.6 Solution Method

The elements of the rectangular grid are sorted into three categories: fully covered by the physical domain, partially covered by the physical domain and totally outside of the physical domain (see figure 40).

We are interested in the first two categories. The treatment of the first category of elements is straightforward and is done in the usual manner.

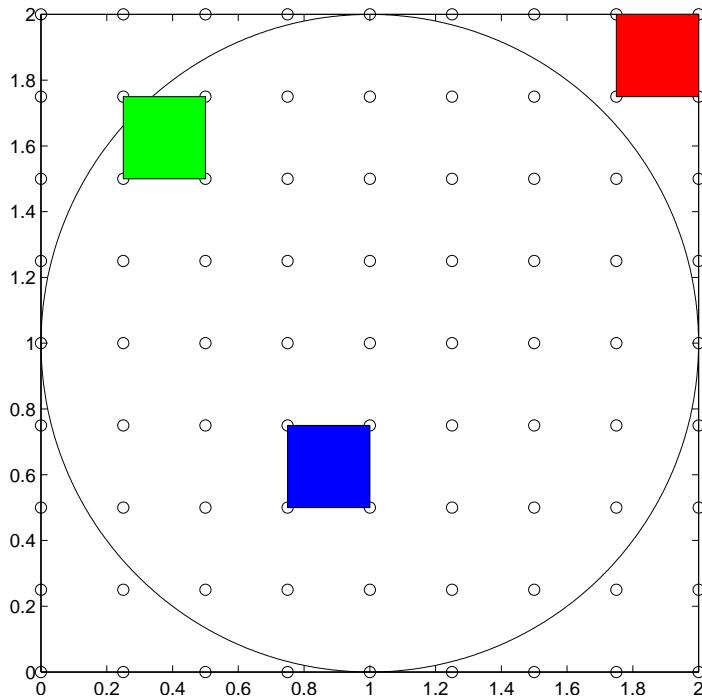


Figure 40: 3 Types of elements: partial (green), full (blue), outside of domain (red)

The partial elements are subject to boundary conditions. In addition, it has to be determined which nodes of the partial elements do lie inside, outside or on the boundary of the domain. The nodes that do lie outside of the domain do no have a material stiffness and therefore have to be treated differently.

Without consideration of the boundary conditions and the hanging nodes of the partial elements, the element stiffness matrices and the element force vectors are generated and assembled for full and partial elements. After assembling, the entries associated with the outside of the domain elements remain zero and have to be taken care of in order to solve the system of equations.

The rows of the system associated with the hanging nodes do contain contribution from the system stiffness which is physically not consistent. However, the hanging nodes are subject to a deformation due to the deformation of the inner nodes as well as the boundary conditions. Therefore, and in analogy to the onedimensional study done before, the row associated with the hanging nodes is substituted by a boundary condition interpolation equation. In order to do this modification, the intersection of the physical domain with the rectangular element has

to be determined. The intersection points could be the hanging nodes themselves, but this is just one special case. The intuition is to interpolate the intersection point condition in terms of the unknowns of the nodes (the unknowns are here scalars). It is optimal to have as many intersection points as hanging nodes or at least more intersection points than hanging nodes. Each intersection point interpolation substitutes an equation in the stiffness matrix. Now, we have to be careful and not substitute the equations randomly. We want at the end to have a non-singular matrix, i.e. zero diagonal entries are not allowed. One has to figure out the appropriate intersection point to choose for an interpolation condition. If the hanging node happens to be an intersection point itself, then we are done. The criteria of having non-zero diagonal elements is a difficult criteria to fulfill once the problems get harder. Therefore, an idea is to take an intersection point lying on one of the edges for which the hanging node is a corner node. This works perfectly smooth assuming correct implementation and treatment of round-off errors.

Now, one has to think of the hanging nodes that either do not coincide with an intersection point and do not have a neighboring intersection point. One can treat this special case by creating “fake” intersection points. These points do lie on the boundary of the physical domain but do not lie on any of the edges of the element. These elements can be created by averaging two intersection points. Averaging might be the wrong word for it, because the average of two intersection points might not lie on the physical domain boundary. Therefore, averaging can happen in one direction (horizontal or vertical) and the associated other coordinate can be determined from the geometric equation of the physical domain. Obviously, the “fake” intersection point does lie inside of the element domain. The interpolation equation does create a diagonal entry.

Some representatives of these cases are depicted in figure 41.

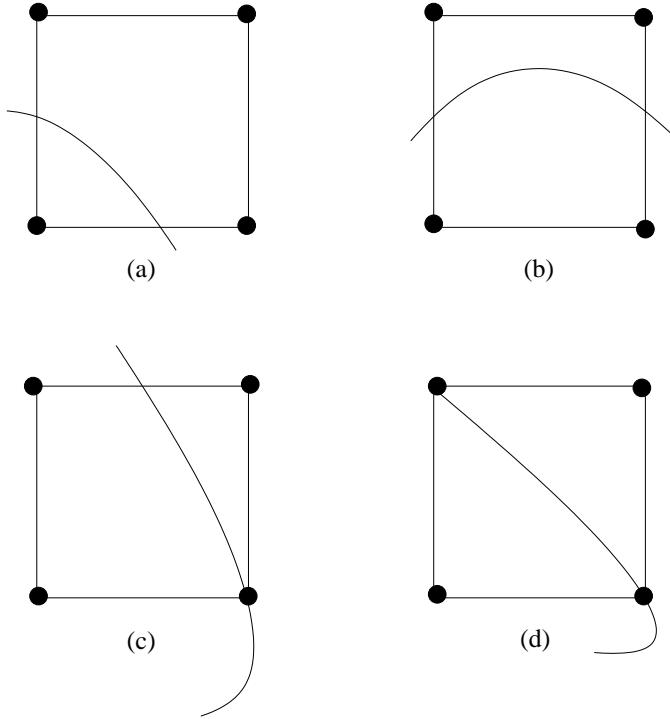


Figure 41: Partial Element Cases: (a) 3 hanging nodes and 2 intersection points (creation of a “fake” intersection point is necessary) (b) 2 hanging nodes and 2 intersection points (c) 2 hanging nodes and two intersection points (one of them is a hanging node) (d) 3 hanging nodes and 2 intersection points (that are 2 hanging nodes)

The stiffness matrix might still contain zero diagonal entries that are associated with nodes coming from elements that do totally lie outside of the domain. One can assume that these nodes do not deform at all. Therefore, the rows and columns containing zero diagonal entries can simply be deleted. From the implementation point of view, an efficient and flexible data storage structure is needed here.

A simplifying assumption has been made here, that an element that lies totally outside of the physical domain except that one node does lie on the boundary, can be treated as being outside of the domain. The “three” hanging nodes are not supposed to deform. The node lying on the boundary will be treated since it belongs to at least one partial element.

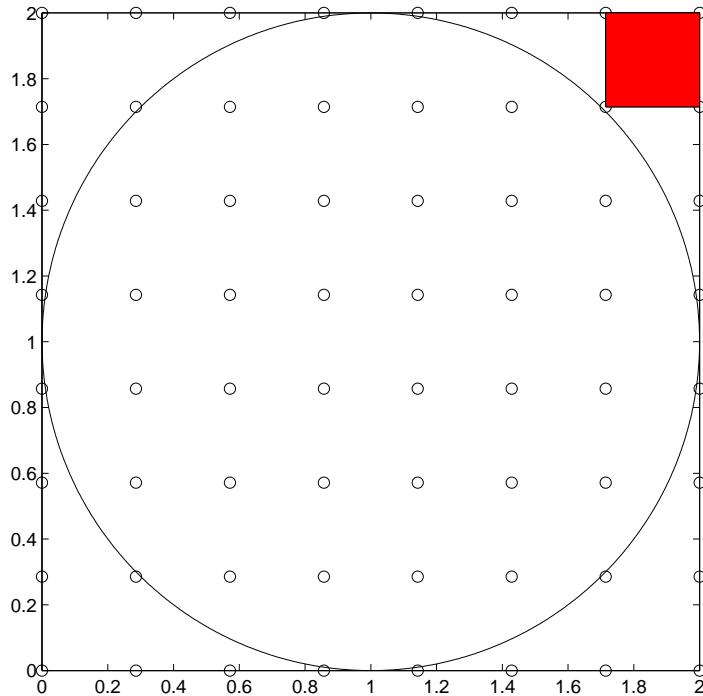


Figure 42: Special Case of Elements Outside of Domain

Just for the sake of completeness, one other special case needs to be handled when the rectangular grid is only covered by a single element. One does take the simple case, where the physical domain is a rectangle, but free shaped. Hence, the element is partially covered by the physical domain and all of the element nodes are hanging nodes (see figure 43). If one does not pay attention, the above discussed algorithm will analyze the element as being outside of the domain since all nodes lie outside of the domain. Therefore, a case catch must be implemented. If the physical domain does not touch the edges of the element, there is no way to solve this problem. But, also if the 4 intersection points do exist and if it is symmetric as shown in figure 43, the interpolation equations will create a rank 0 system that is unsolvable. But, seriously, we do not expect an accuracy from solving a complicated finite element problem using a single element. Therefore, one might just want to omit this case and consider solving this problem using let's say a minimum of 4 elements.

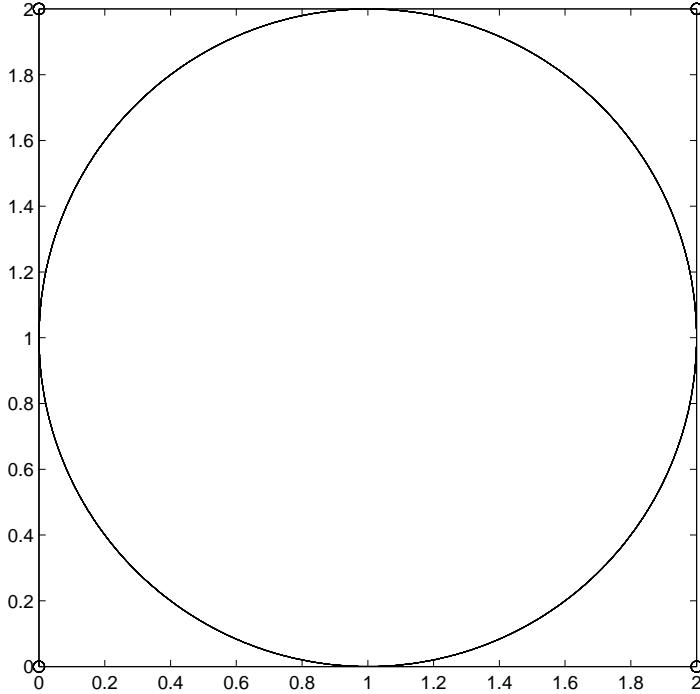


Figure 43: One Element Rectangular Grid

7.7 Convergence Study: First Attempt

We want to study and understand the convergence of this new algorithm. We solve the following problem:

$$A \Delta u + f = 0, \quad (164)$$

$$A = 1, \quad f = 1, \quad (165)$$

$$u = \bar{u} = 0.0 \quad \text{on the entire boundary.} \quad (166)$$

For different rectangular grid element sizes, the error that is made by this algorithm to the interior node values is determined as follows:

$$\|e\| = \frac{\|\mathbf{u}_n^i - \mathbf{u}^i\|_\infty}{n}, \quad (167)$$

$$= \frac{\sum_j |\mathbf{u}_{n_j}^i - \mathbf{u}_j^i|}{n}, \quad (168)$$

where n is the total number of the interior nodes to the physical domain, \mathbf{u}_n^i is the algorithm solution at the interior nodes and \mathbf{u}^i is the analytical solution at the nodes. The analytical solution has been derived in section 7.2. The error as a function of the total number of interior nodes is depicted in the log-log-plot in figure 44. One notices the consistent linear convergence of the algorithm, i.e. the error is of the order of the element size. One does expect

this behavior because of the non-propagation of the boundary information to the neighboring elements. The method imposed the boundary condition at the hanging nodes using an interpolation equation in the domain of one element. We suspect that the expansion of the interpolation domain will effect the convergence behavior positively.

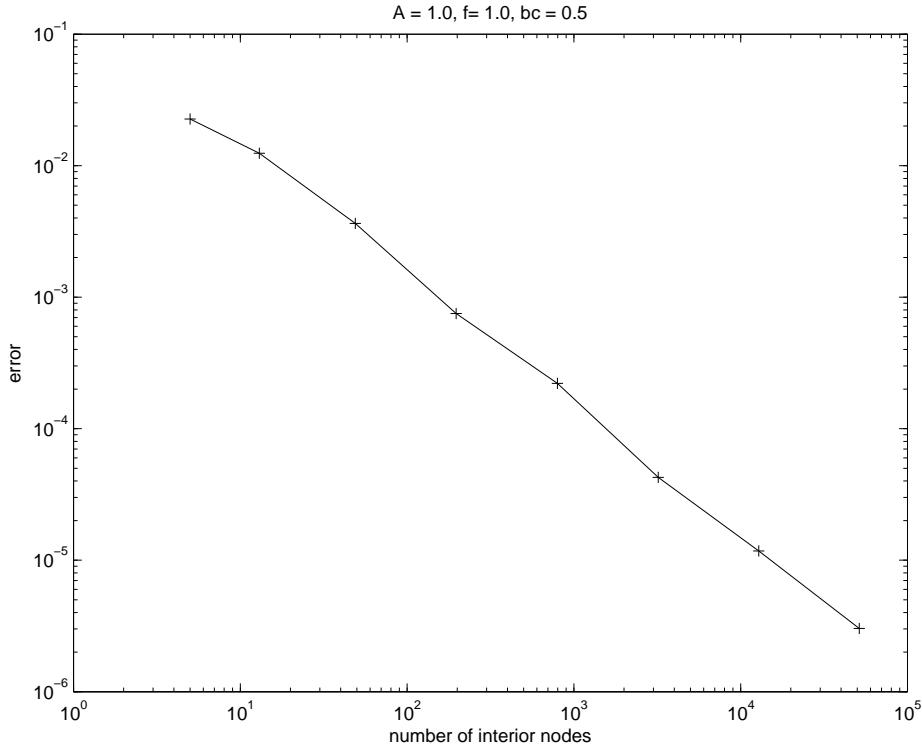


Figure 44: Convergence Study

The test results are tabulated in the table 1:

$\ e\ $	n
0.022613849221600	5
0.012440258770754	13
0.003636812889227	49
7.512968349467671e-04	197
2.211819084109193e-04	797
4.266346728977254e-05	3209
1.173832783656672e-05	12853
3.024078476989026e-06	51433

Table 1: Convergence Test Results

There is a small undesirable kink in the middle of the convergence curve. We correct this by doing an additional test. The corrected convergence behavior is depicted in figure 45 and the additional test results are given in table 2.

$\ e\ $	n
1.527300058654243e-04	848
1.160100600789700e-04	1257

Table 2: Additional Convergence Test Results

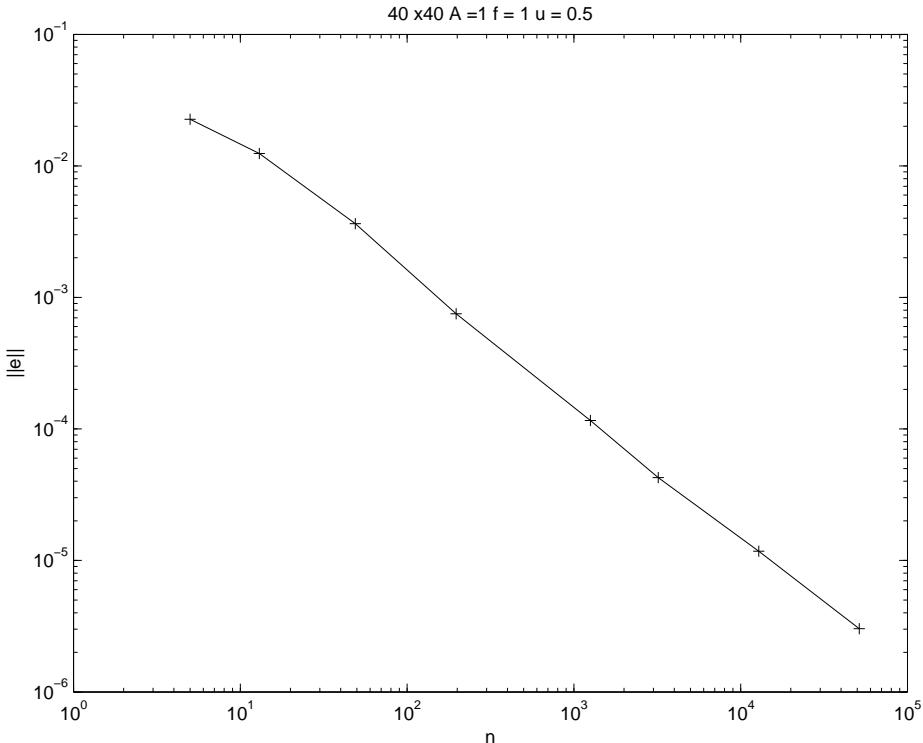


Figure 45: Corrected Convergence Study

As we see, the curve is not really straight. Actually, when we further refine the mesh, the curve loses linearity and drops vertically to a smaller error. One wonders about the reasons of this behavior. A thought would be that elements with very small physical coverage ratio are present and affect the presented algorithm. A second look at the convergence behavior made us rethink about our statement. The H0 error norm we are introducing in equation (167) should be quadratic in h, i.e. the slope in the log-log plot of the error as a function of the number of interior nodes should be of order 2. The choice of the number of the interior node as the independent variables turns out to be a bad choice. In the following sections, we will see that the independent variable should be better the element size.

The more we look at the numerical solution and the analytical solution and compare them, we doubt on the convergence conclusions we made above. The two solutions are depicted in the figures 46 and 47.

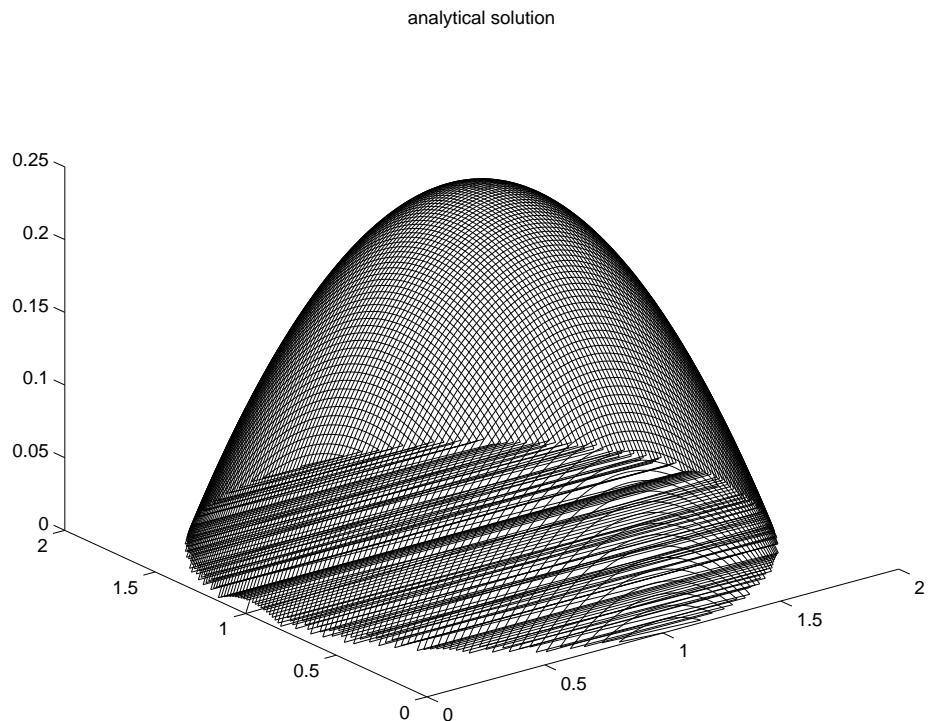


Figure 46: Analytical Solution for Circular Domain

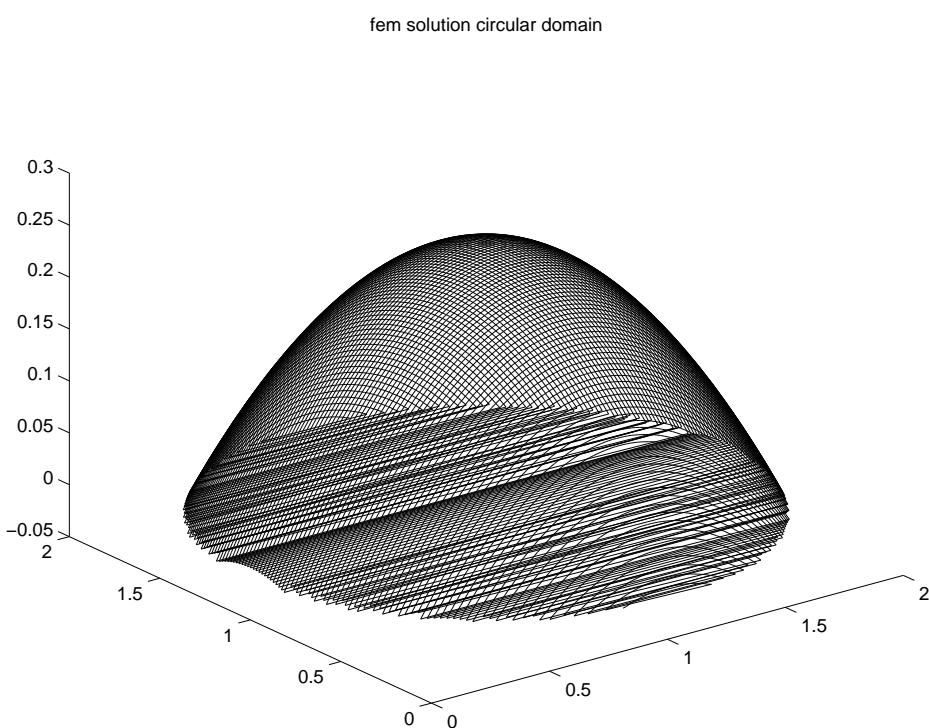


Figure 47: Numerical Solution for Circular Domain

7.8 Comparison to the One-Dimensional Algorithm

We want to verify the convergence behavior of the two dimensional algorithm by comparing it to the convergence behavior of the one-dimensional problem. We studied the one-dimensional problem carefully, and we were able to make consistent conclusions.

We will solve the one-dimensional problem given in (1) for different mesh sizes, with the following parameters:

$$\bar{u}|_0 = 0.0, \quad \bar{u}|_{L_c} = 0.5, \quad K = -1 \quad (169)$$

$$A = 1, \quad L_c = 1, \quad L_0 = 0, \quad (170)$$

where L_0 and L_c are the left and right coordinates of the one-dimensional physical domain. We follow the error analysis and definition presented in section 7.7. Using a linear interpolation, i.e no coupling between the elements, the test results are tabulated in the table 3 and depicted in the log-log-plot 48:

$\ e\ $	n
0.031250000000000	2
0.007812500000000	4
0.001953125000000	8
4.88281250000000e-04	16
1.220703125001110e-04	32
3.051757812500000e-05	64
7.629394531250000e-06	128
1.907348632701478e-06	256
4.768371582031250e-07	512
1.192092895507812e-07	1024
2.980232238769531e-08	2048
7.450580596923828e-09	4096

Table 3: Convergence Test Results of One-dimensional Case Without Coupling

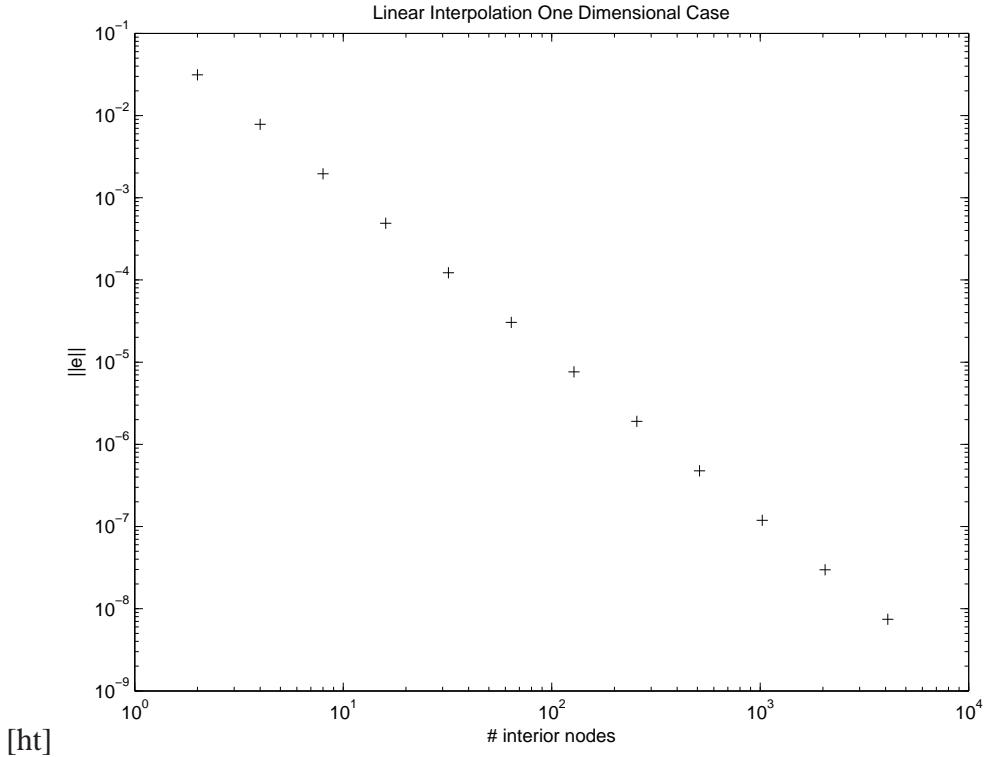


Figure 48: Convergence Study of One-dimensional Case Without Coupling

The results obtained show quadratic convergence that does not confirm the results obtained from the two-dimensional counter-example. The error is calculated at the nodes that are considered to be superconvergent, hence the quadratic convergence. Therefore, something is wrong with the error analysis for the two-dimensional case.

For the one-dimensional case, if we propagate the interpolation information to a neighbor element, i.e. if we couple the two first and the two last elements, the one-dimensional study does lead to an interesting effect. An error of the order of the machine epsilon is made, when one would use only two elements. I.e. the coupling hits the exact solution for the coarsest possible element size. Decreasing the mesh size leads to an increase in the error, even though the numerical solution does match the analytical one (Results are tabulated in table 4). We suspect that the round-off errors overwhelm the numerical error. Since the numerical error is tiny small, if not vanishing for the problem we are solving, the round-off error dominates and leads to an increase in the total error instead of a decrease.

$\ e\ $	n
4.163336342344337e-17	2
1.296705798292663e-16	4
1.112825109839122e-15	8
2.465150479580291e-15	16
2.081267016368349e-14	32
4.223409274216328e-14	64
3.404037170412835e-13	128
6.676058021833596e-13	256
5.472485393445345e-12	512
1.088767981445528e-11	1024
8.732092397230042e-11	2048
1.748866421536470e-10	4096
1.396787000525363e-09	8192

Table 4: Convergence Test Results With Coupling

If we look at the numerical solution u_n and compare it to the analytical solution u for different element sizes, we deduce that the roundoff error is overwhelming the truncation error. The algorithm hits the analytical solution almost exactly and due to the nature of the problem (linear) but the numerical noise leads to an increase in the error. See tables 5, 6 and 7.

u_n	u
0.031250000000000	0.031250000000000
0.031250000000000	0.031250000000000
0.281250000000000	0.281250000000000
0.781250000000000	0.781250000000000

Table 5: Comparison of Analytical and Numerical Solution for $h = 0.5$

u_n	u
0.007812500000000	0.007812500000000
0.007812500000000	0.007812500000000
0.070312500000000	0.070312500000000
0.195312500000000	0.195312500000000
0.382812500000000	0.382812500000000
0.632812500000000	0.632812500000000

Table 6: Comparison of Analytical and Numerical Solution for $h = 0.25$

u_n	u
0.001953125000000	0.001953125000000
0.001953125000000	0.001953125000000
0.017578125000000	0.017578125000000
0.048828124999999	0.048828125000000
0.095703124999999	0.095703125000000
0.158203124999999	0.158203125000000
0.236328124999998	0.236328125000000
0.330078124999998	0.330078125000000
0.439453124999998	0.439453125000000
0.564453124999998	0.564453125000000

Table 7: Comparison of Analytical and Numerical Solution for $h = 0.125$

7.8.1 Eigenvalue-Analysis for the Quadratic Interpolation Case

In order to approve or disprove our suspicion, we want to conduct an eigenvalue analysis of the system we are solving in the one-dimensional case when we use the two-elements-coupling technique.

We first want to see that the system is well-conditioned before inserting the interpolation equation for imposing the boundary equations and to make sure that the ill-conditioning arises after imposing the boundary conditions.

Effectively, the condition number of the system (symmetric tridiagonal system matrix), $\kappa(\mathbf{K})$, does not vary with the element size h . It is however high as it was expected because it is singular (see table 8).

$\ \kappa(\mathbf{K})\ $	h
0.5	4.085925719606174e+16
0.25	1.288222685639135e+16
0.1250	1.102518255522341e+16
0.0625	1.610545333263965e+17
0.03125	4.216490311080406e+16
0.015625	2.842184018423543e+16
0.0078125	2.849914728689063e+16
0.00390625	2.069591063438704e+16
0.001953125	9.754538101110034e+16
9.76562500000000e-04	1.384525765787646e+16

Table 8: Condition Number of the System Matrix For Different Element Sizes Before Imposing Boundary Conditions

It is obvious that the system matrix loses symmetry when we substitute the equation associated with the first and last degrees of freedom with the interpolation equations. The substitution of the equations is simply a subtraction of a rank two matrix and a consecutive

addition of a rank two matrix.

Instead of looking at the condition number separately, it is more appropriate to look at the error as a function of the condition number of the final system, i.e. after inserting the interpolation equations.

For perturbed systems, we know that the relative⁷ perturbation error is correlated to the condition number of the system and the machine epsilon by the following relationship:

$$\|e\| = \varepsilon \kappa(\mathbf{K}), \quad (171)$$

$$\log(\|e\|) = \log(\kappa(\mathbf{K})) - 16, \quad (172)$$

for double precision calculations.

If we can show that this relationship holds for our error, then we can confirm that the error that is happening is due to round-off solely. See figure 49.

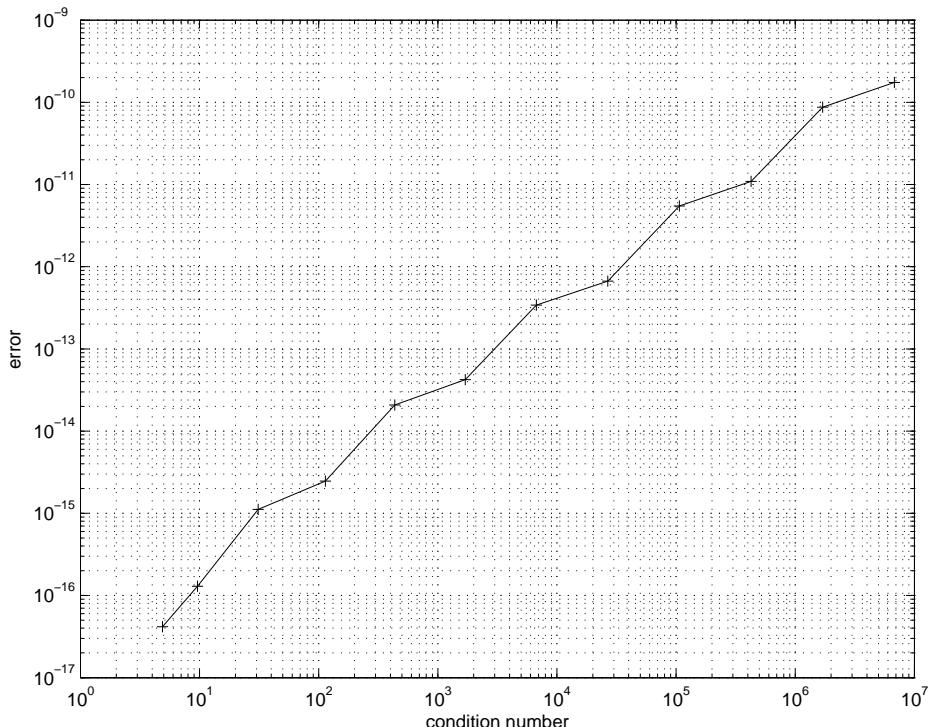


Figure 49: Correlation of the error and the condition number of the system

The graph shows that the total error is linear dependent on the condition number of the system matrix, and we can maybe conclude that the total error includes only round-off errors.

The graph depicted in 50 confirms that the total error is exclusively the round-off error that is bounded by the product of the machine epsilon and the system condition number.

⁷The studies in this section has been made accidentally using the absolute error which happens to be the same as the relative error. This is due to the fact that the finite element solution vector has a unit length.

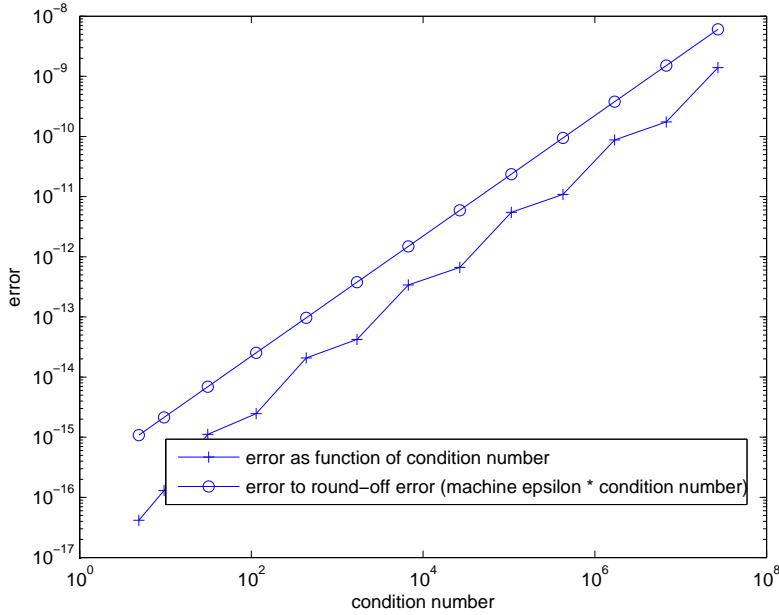


Figure 50: Correlation of the error and the condition number of the system vs. Round-off Error Upper Bound

A way to deal around the round-off error is to whether increase the precision of the variables, but the matlab version the code has been written in does not allow more. An interface to a programming language with higher precision possibilities can be used, but was not explored at this stage.

Also, a study of a nonlinear solution by introducing source terms into the system of equations and increasing the stiffness of the physical domain will work around the round-off errors. Since the finite element solution uses linear elements, the nonlinear behavior will never be accurately reproduced. Therefore, the truncation error will dominate the round-off error, and a convergence study will show a decrease in the overall error for finer meshes.

The latter solution has been tried but the finite element solution seems to be very close to the analytical solution. The roundoff error still dominates. A solution of a highly nonlinear problem is required to kill off the round-off error. Therefore, instead of solving the problem:

$$Au_{xx} + K = 0, \quad (173)$$

where both A and K are constants, we solve:

$$Au_{xx} + x = 0, \quad (174)$$

where linear source terms are introduced to the partial differential equation. Linear finite elements would never be able to reproduce the cubic shape of the solution. Hence, the truncation error should be existing and hopefully dominates the round-off error. The convergence study would allow us then to interpret the performance of both the linear interpolation and

the quadratic interpolation schemes used to impose the boundary conditions appropriately. Figures 51 and 52 show that the linear interpolation technique to impose the Dirichlet boundary condition leads to a poor accuracy and slow convergence whereas the quadratic interpolation scheme makes a smaller error. The Neumann boundary condition is imposed using the derivatives of the interpolation equations used to impose the Dirichlet boundary condition. I.e. linear and midpoint interpolation techniques are used. Here also, the higher order interpolation technique leads to a slightly better performance. Intuitively, a quadratic interpolation technique would be a better way to impose the Neumann boundary condition.

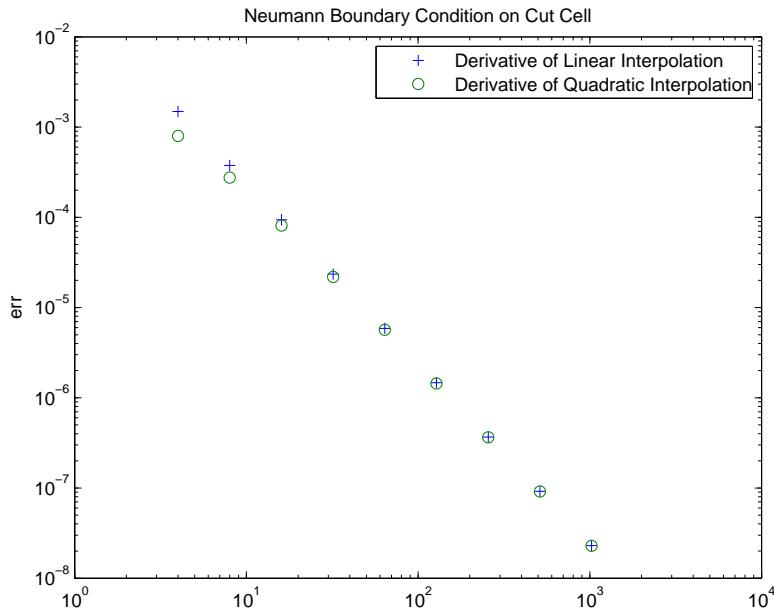


Figure 51: Neumann Problem: Derivative of Linear Interpolation vs. Derivative of Quadratic Interpolation

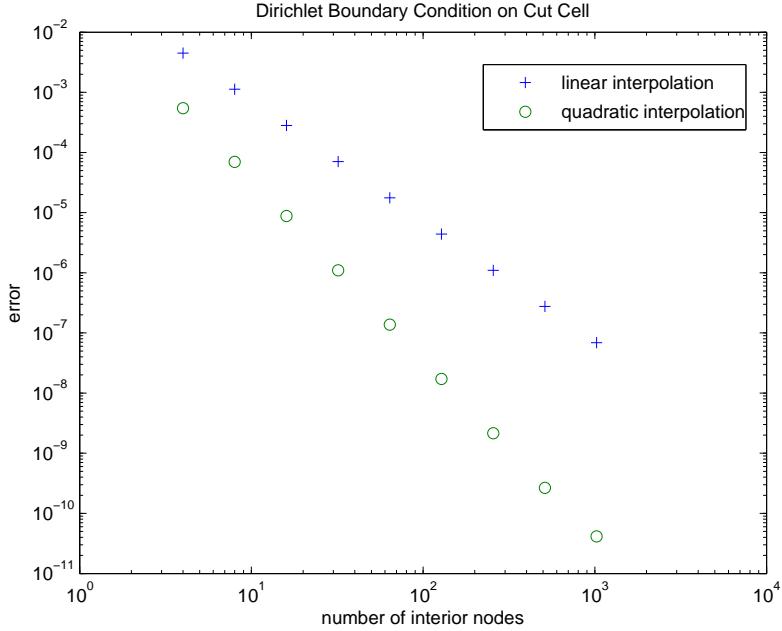


Figure 52: Dirichlet Problem: Linear Interpolation vs. Quadratic Interpolation

There exists a method that targets the round-off error, the iterative refinement technique. One applies the Newton-method to the linear system, even if convergence is going to be obtained in a single step. The round-off error is killed (see [2] for more information about this technique). Existing implementation to this technique is provided in LAPack.

7.8.2 Different Error Norms and Independent Variables

Since we obtained 'just' linear convergence in the two-dimensional case, but a quadratic convergence for the analogous one-dimensional case, one is not sure about the cause of this difference. One thought would be that the choice of the norm to show convergence is not an appropriate one. In this section, we want to investigate the effect of the choice of the error norm on the convergence behavior. Until now, we have used an approximation for the H0-Norm as defined in (167). We want to expand the error analysis to the use of the "exact" H0-Norm:

$$\|e\|^2 = \int \int (u - u_n)^2 dx dy, \quad (175)$$

where u is the analytical solution and u_n is the numerical solution.

We also want to use the H1-Norm defined as follows:

$$\|e\|^2 = \int \int (u - u_n)^2 dx dy + \int \int \left[\left(\frac{\partial(u - u_n)}{\partial x} \right)^2 + \left(\frac{\partial(u - u_n)}{\partial y} \right)^2 \right] dx dy, \quad (176)$$

and the energy norm as in:

$$\|e\|^2 = \int \int \left[A \left(\frac{\partial(u - u_n)}{\partial x} \right)^2 + A \left(\frac{\partial(u - u_n)}{\partial y} \right)^2 \right] dx dy, \quad (177)$$

where A is the material parameter used in the differential equation. Using these norms, an error analysis has been made. It might be 'cheating', but it is appropriate to discuss the choice of the independent variable against which we evaluate the convergence behavior. For the approximated norm used in section 7.7, we used the number of the interior nodes as the independent variable. Another idea is to use to the total number of elements in one of the directions as an independent variable. The results are depicted in figures 53, 54, 55, 56.

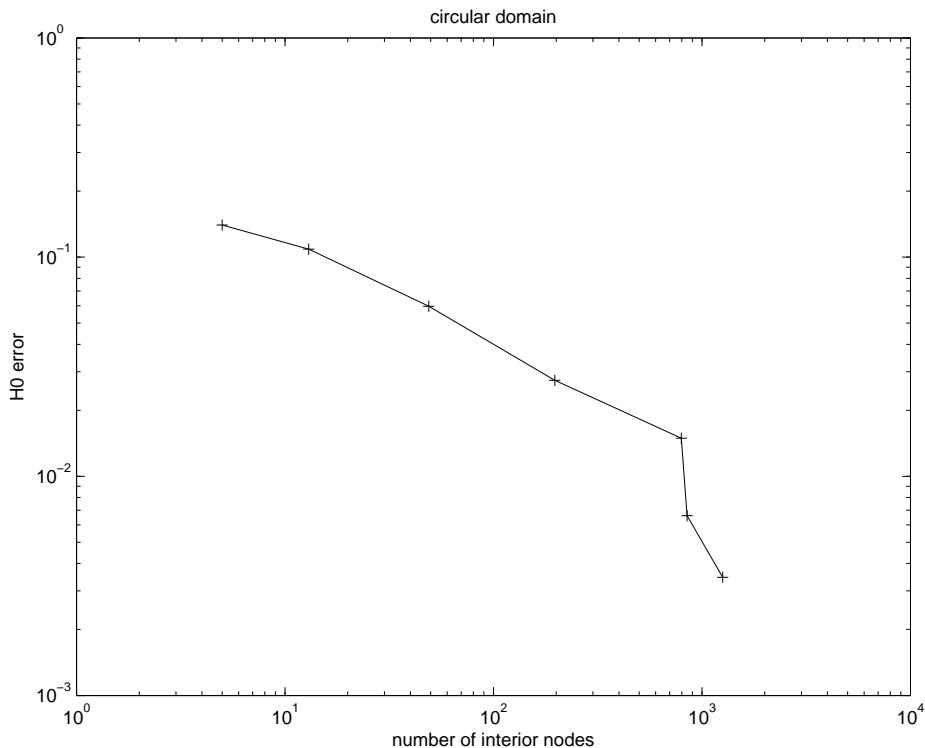


Figure 53: H0 error norm vs. the number of interior nodes

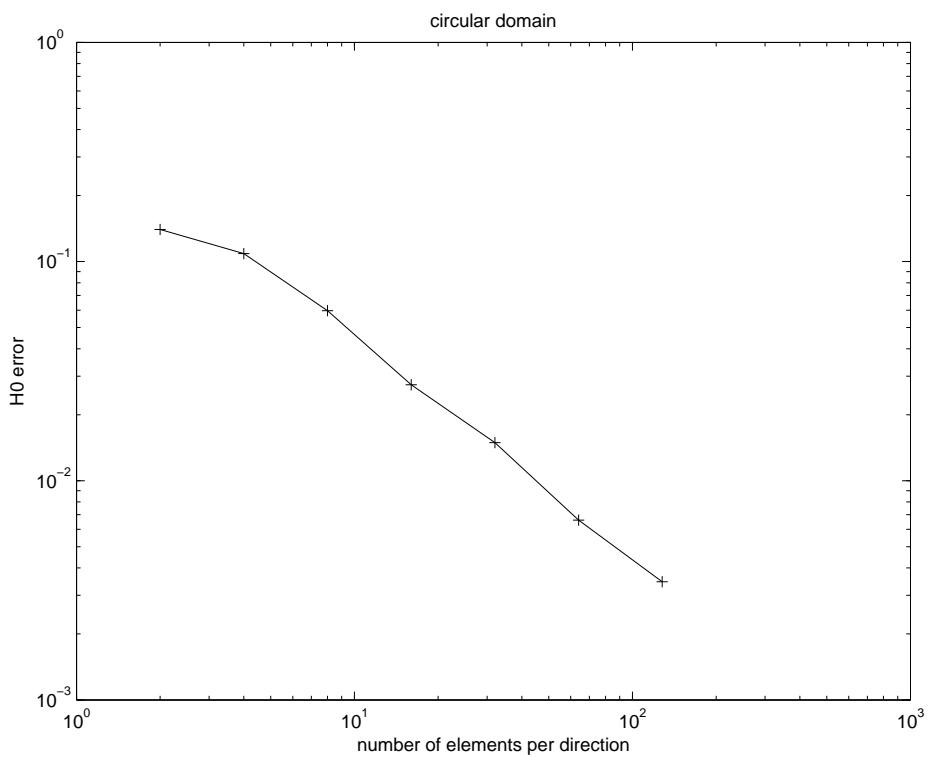


Figure 54: H_0 error norm vs. the number of elements in one-direction

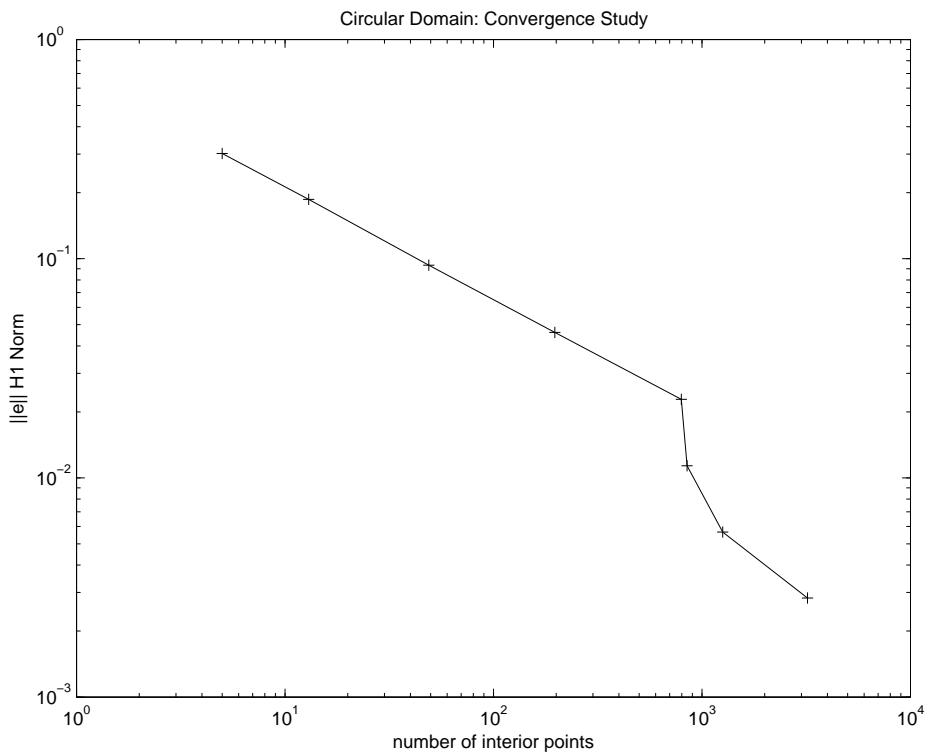


Figure 55: H1 error norm vs. the number of interior nodes

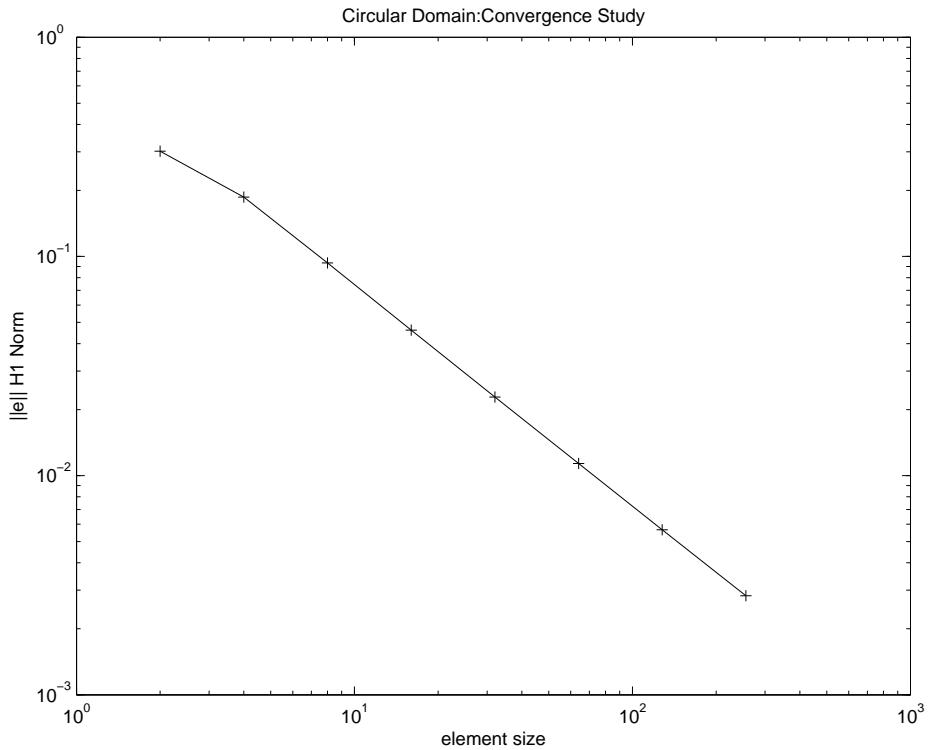


Figure 56: H1 error norm vs. the number of elements in one-direction (the xlabel should read 'the number of elements in one-direction'

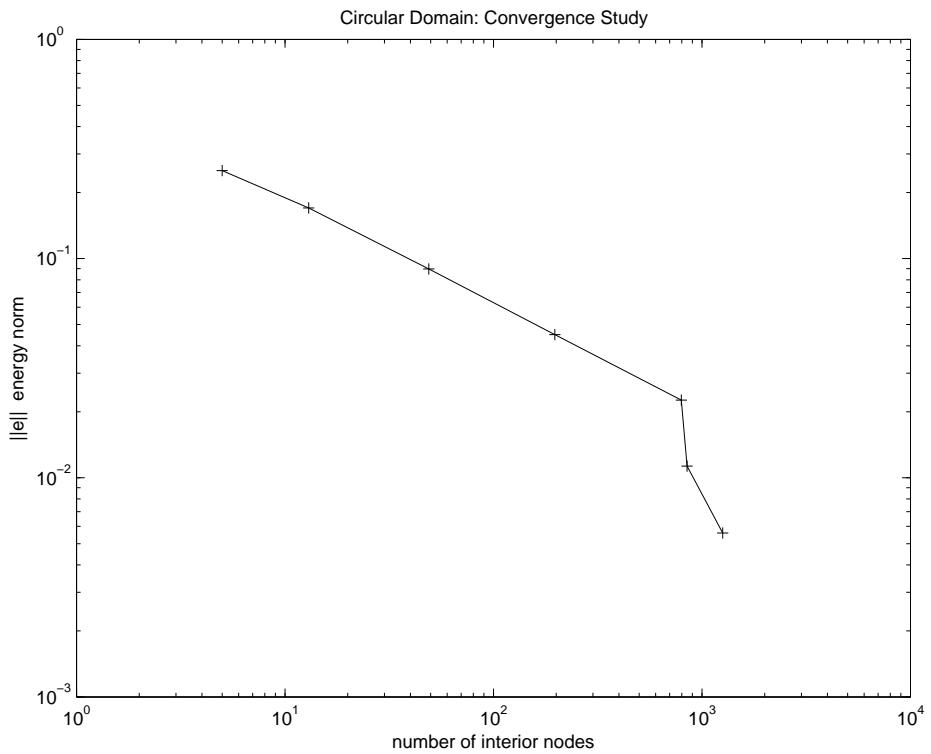


Figure 57: Energy error norm vs. the number of interior nodes

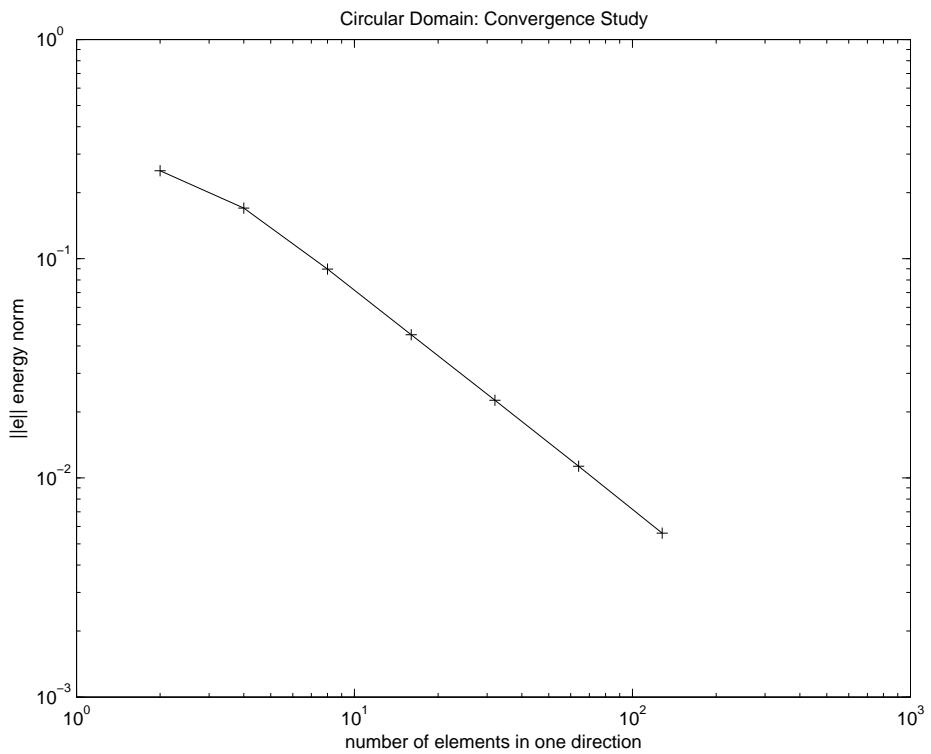


Figure 58: Energy error norm vs. the number of elements in one-direction

We obtained two different behaviors using the two different independent variables, the number of interior nodes and the number of elements per direction. The correlation between the two independent variables is given in figure 59 and explains the difference in convergence.

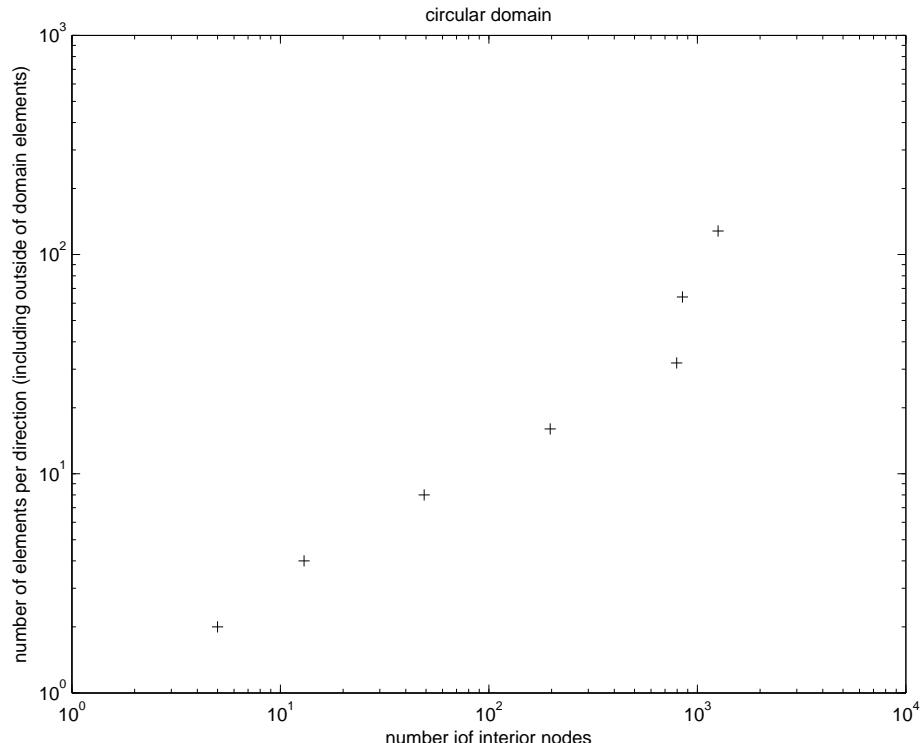


Figure 59: Correlation between the number of interior nodes and the number of elements per direction

7.9 Different Geometry: Rectangular Domain

In order to make sure, that the two-dimensional case and the analogon one-dimensional case do exhibit the same convergence behavior, we solve the canonical problem discussed in section 7.1 on a rectangular domain. This would change the analytical solution as well as the numerical solution algorithm discussed above, since all elements collapse into one single category: full elements. The solution algorithm acts therefore as regular finite elements without the notion of partial elements or ghost nodes. The rectangular region is to be defined on the domain $\{0 \leq x \leq a, 0 \leq y \leq b\}$.

7.9.1 Analytical Solution: Wrong One

We want to solve:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{f}{A}, \quad (178)$$

$$u(x, 0) = \bar{u}, \quad (179)$$

$$u(x, b) = \bar{u}, \quad (180)$$

$$u(0, y) = \bar{u}, \quad (181)$$

$$u(a, y) = \bar{u}. \quad (182)$$

The homogeneous solution to this problem reads:

$$u_h(x, y) = \sum_{n=1}^{\infty} A_n \sinh\left(\frac{n\pi}{b}(a-x)\right) \sin\left(\frac{n\pi}{b}y\right) + \sum_{n=1}^{\infty} A_n \sinh\left(\frac{n\pi}{b}x\right) \sin\left(\frac{n\pi}{b}y\right) \quad (183)$$

$$+ \sum_{n=1}^{\infty} B_n \sinh\left(\frac{n\pi}{a}(b-y)\right) \sin\left(\frac{n\pi}{a}x\right) + \sum_{n=1}^{\infty} B_n \sinh\left(\frac{n\pi}{a}y\right) \sin\left(\frac{n\pi}{a}x\right), \quad (184)$$

$$A_n = \frac{2b}{n\pi\lambda_n} \bar{u} (1 - \cos(n\pi)), \quad (185)$$

$$B_n = \frac{2a}{n\pi\mu_n} \bar{u} (1 - \cos(n\pi)), \quad (186)$$

$$\lambda_n = b \sinh\left(\frac{n\pi a}{b}\right), \quad (187)$$

$$\mu_n = a \sinh\left(\frac{n\pi b}{a}\right). \quad (188)$$

Since the homogeneous solution did satisfy the boundary conditions, the particular solution is subject to the following boundary conditions:

$$u_p(x, 0) = 0, \quad (189)$$

$$u_p(x, b) = 0, \quad (190)$$

$$u_p(0, y) = 0, \quad (191)$$

$$u_p(a, y) = 0. \quad (192)$$

The particular solution should satisfy the right hand side of the differential equation, since the homogeneous solution vanishes after differentiation. **No appropriate particular solution has been found.**

7.9.2 Convergence Analysis

For the homogeneous Laplace equation on the rectangular domain, the approximated H^0 -error made between the analytical and the numerical solution for n interior nodes is given in figure 60. The values are tabulated in table 9.

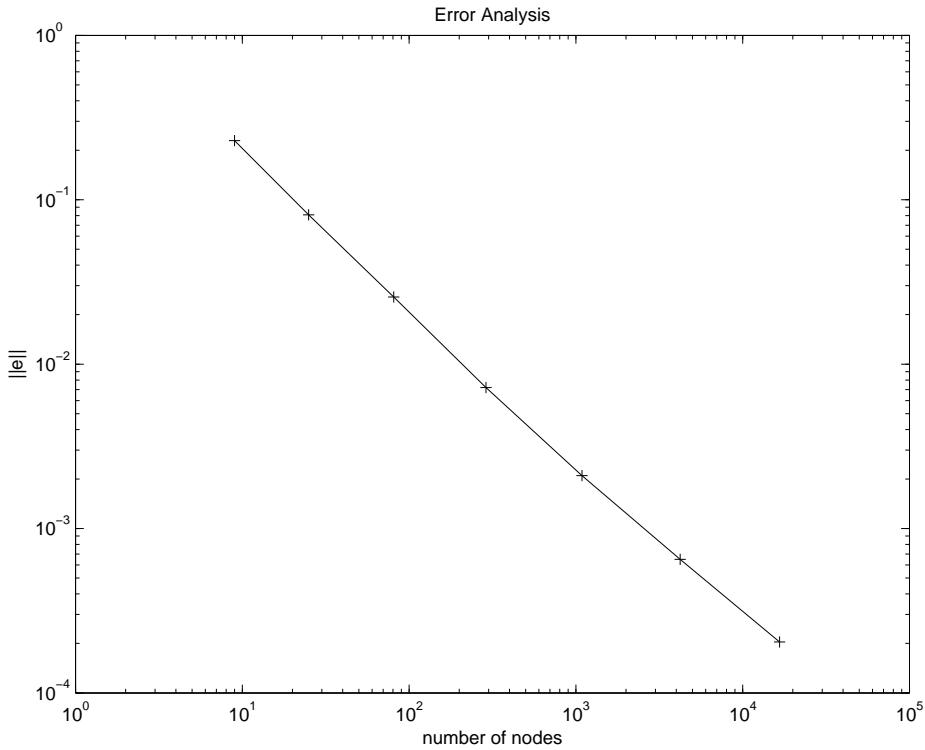


Figure 60: Rectangular Domain: Convergence Study

$\ e\ $	n
0.229	9
0.081	25
0.0256	81
0.0072	289
0.0021	1089
6.4985e-04	4225
2.0430e-04	16641

Table 9: Rectangular Domain: Convergence Test Results

It is frustrating that it does still show linear convergence, even for the regular geometry and method case.

7.9.3 Analytical Solution: Corrected

It appeared that the analytical solution to the problem stated in (178) can be obtained by means of Green's functions and reducing the differential equation to an integral equation. The Green's function for the Laplacian in the interior of a rectangular domain corresponding

to the boundary condition $\bar{u} = 0$:

$$K(x, y; \xi, \eta) = \frac{4}{ab\pi^2} \sum_{m=1}^{\infty} \sum_{k=1}^{\infty} \frac{\sin(k\frac{\pi}{a}x)\sin(m\frac{\pi}{b}y)\sin(k\frac{\pi}{a}\xi)\sin(m\frac{\pi}{b}\eta)}{\frac{m^2}{b^2} + \frac{k^2}{a^2}}. \quad (193)$$

The solution to the problem reads therefore:

$$u(x, y) = \int_0^b \int_0^a K(x, y; \xi, \eta) \frac{f}{A} d\xi d\eta, \quad (194)$$

which can be integrated numerically using Gaussian quadrature. Moreover, a series can never be realized numerically. Therefore, the series in equation 193 is approximated by a sum running from 1 to 200 which is tested to be accurate enough in double precision.

The convergence behavior of the numerical solution in the H0-, H1- and energy-norms as a function of the mesh size is given in figure 61. The convergence rate is quadratic as a function of the mesh size if the error made with respect to the analytical solution is given in the H0-norm. Both the H1- and the energy error norms exhibit a linear convergence rate. This convergence study gives us the expected convergence rates from the new developed algorithm using the partial cells concept.

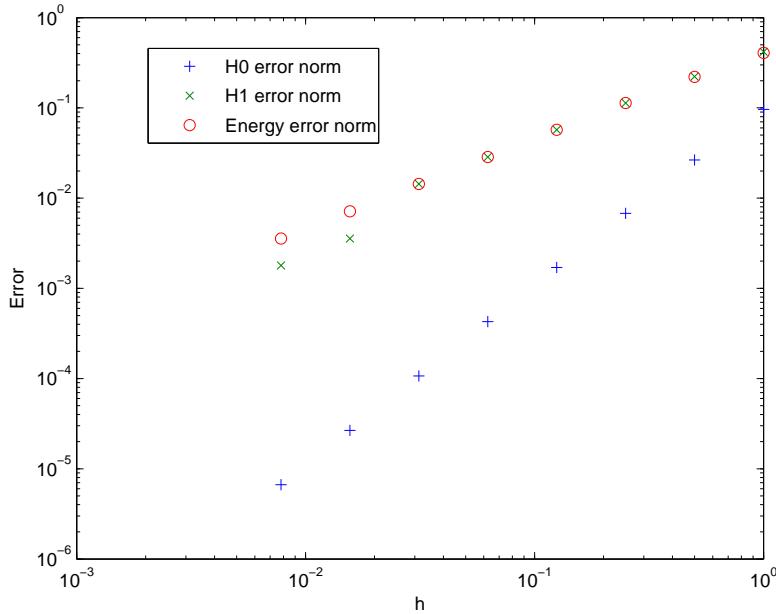


Figure 61: Rectangular Domain: Convergence Study

8 Circular Domain: Convergence Study the 2nd

As cited in [1], the mesh nodes are superconvergent points and should exhibit a high convergence rate in the one-dimensional case at least. The nature of the two-dimensional problem

constitutes a family of one-dimensional problems. Therefore, one might expect superconvergence in the two-dimensional case as well. Moreover, the convergence study made in the two-dimensional rectangular case has promised quadratic convergence in the H_0 -norm and a linear convergence in the H_1 and in the energy norm. This confirms the results we obtained in the one-dimensional case and suggests checking the partial cells code in the two-dimensional case.

The combination of the error norms and the independent variables, with respect to which convergence is studied, used to analyse the convergence rate and investigated in section 7.8.2 are effectively not appropriate. The error norms have been changed to be exact numerically and not an approximation. This issue is fixed. But the independent variables, the number of interior nodes $\mathcal{O}(n^2)$ – or the number of elements in one direction $\mathcal{O}(n)$ –, do not seem to relate correctly to the proposed norms. A different independent variable must be chosen.

Another thought is that the drop in the convergence rate might be due to the interpolation scheme for enforcing the Dirichlet boundary conditions. For every intersection point of the domain boundary with the rectangular grid, that is also used as an interpolation point, an equation is added to substitute an existing one in the system for a particular chosen degree of freedom. Bilinear interpolation in the element to which the intersection point belongs simplifies in every case to involve only two nodes of the bilinear element (Figure 62).

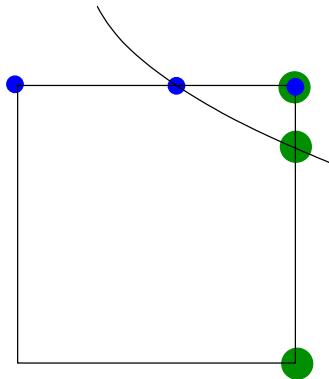


Figure 62: Each Intersection Point Involves Two Element Nodes

Information about the boundary conditions is better propagated if all four element nodes are involved in the bilinear interpolation, i.e. a bidirectional bilinear interpolation must be enforced. Intersection points with the rectangular grid can not be used as interpolation points and need therefore to be modified (section 8.2).

8.1 New Independent Variable: Element Width

The element width h seems to be a better parameter to which the error can be related. The convergence study is consistent and does not exhibit an inexplicable behavior as in section 7.8.2. Analogously to the one-dimensional case, the error given in the H_0 -norm relates quadratically to the element width (Figure 63).

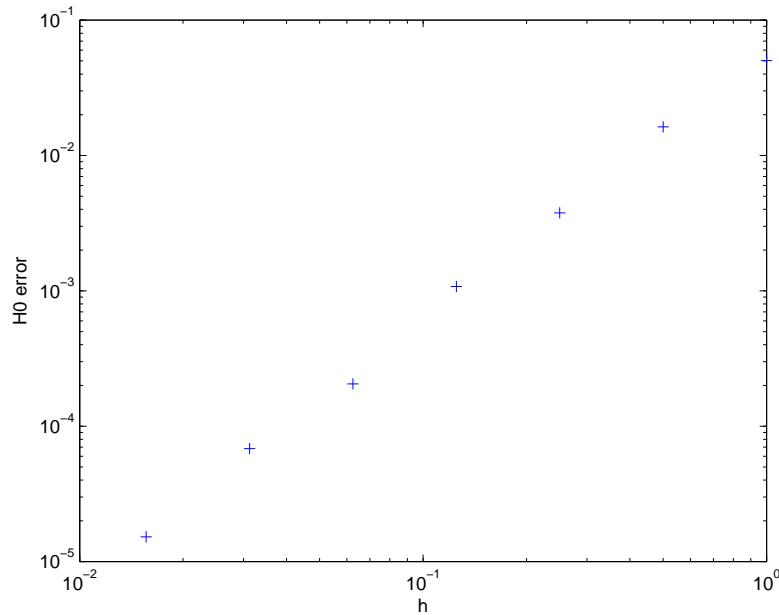


Figure 63: Convergence rate in the H_0 -error norm

The error in the H_1 and in the energy norm do relate linearly to the element width (Figures 64 and 65). This convergence rate are comparable to the results obtained for the convergence study of a simple rectangular domain, where all elements do fully belong to the physical domain.

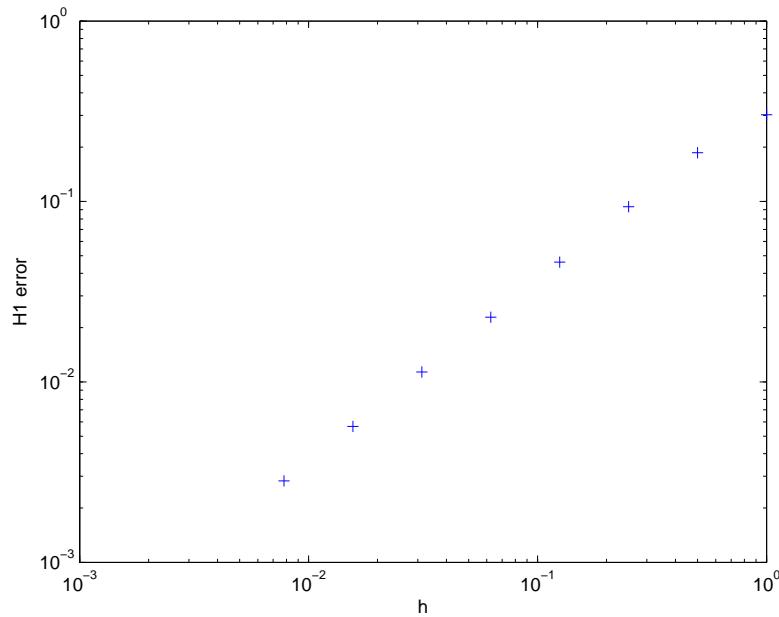


Figure 64: Convergence rate in the H_1 -error norm

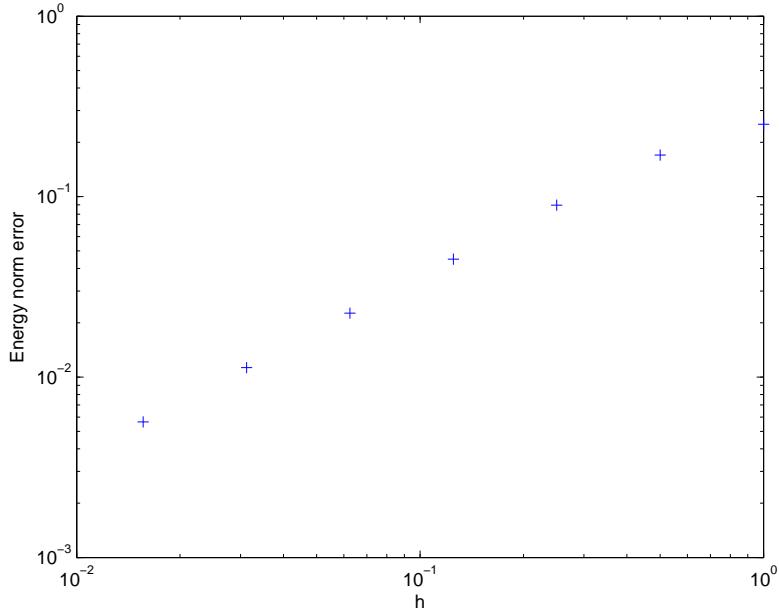


Figure 65: Convergence rate in the energy norm

8.2 Bidirectional Interpolation vs. Simple Interpolation

A simple idea to modify the interpolation points starting from the intersection points of the domain boundary with the rectangular grid is adopted.

Assuming there do exist two intersection points of the physical domain with a particular element:

$$S_1 = (x_1, y_1), \quad (195)$$

$$S_2 = (x_2, y_2), \quad (196)$$

and assuming that three hanging nodes, nodes outside of the physical domain boundary, do exist. Three interpolation equations have to be created, and they should involve all four nodes.

The interpolation points read:

$$I_1 = (0.5\{x_1 + 0.5(x_1 + x_2)\}, y_{I1}), \quad (197)$$

$$I_2 = (0.5\{x_2 + 0.5(x_1 + x_2)\}, y_{I2}), \quad (198)$$

$$I_3 = (0.5(x_1 + x_2), y_{I3}), \quad (199)$$

where y_{I1} , y_{I2} and y_{I3} are the corresponding y -coordinates obtained from the geometrical equation of the physical domain boundary, in this case the equation of a circle. Readily, all interpolation nodes involve all element nodes.

The convergence behavior of the partial cells finite element solution in different norms using

the bidirectional interpolation technique is compared to the simple interpolation technique (Figures 66, 67 and 68).

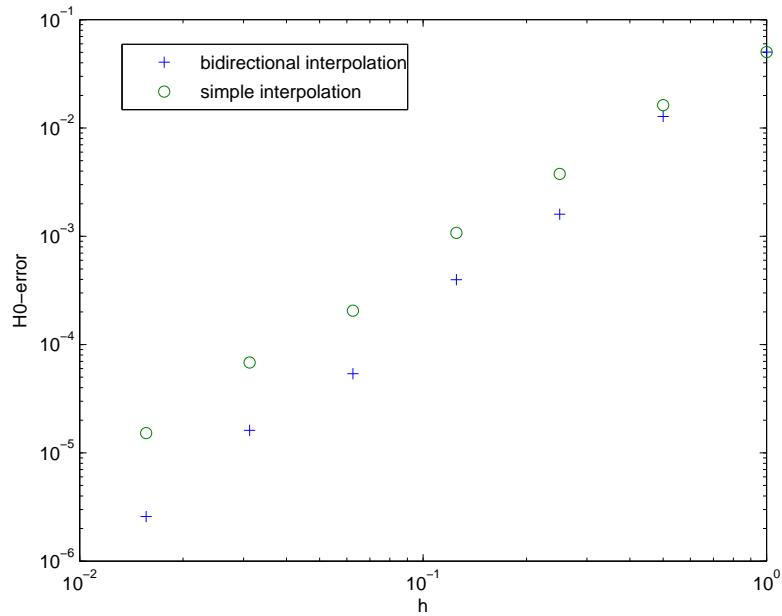


Figure 66: Convergence rate for bidirectional and simple interpolation in the H0-error norm

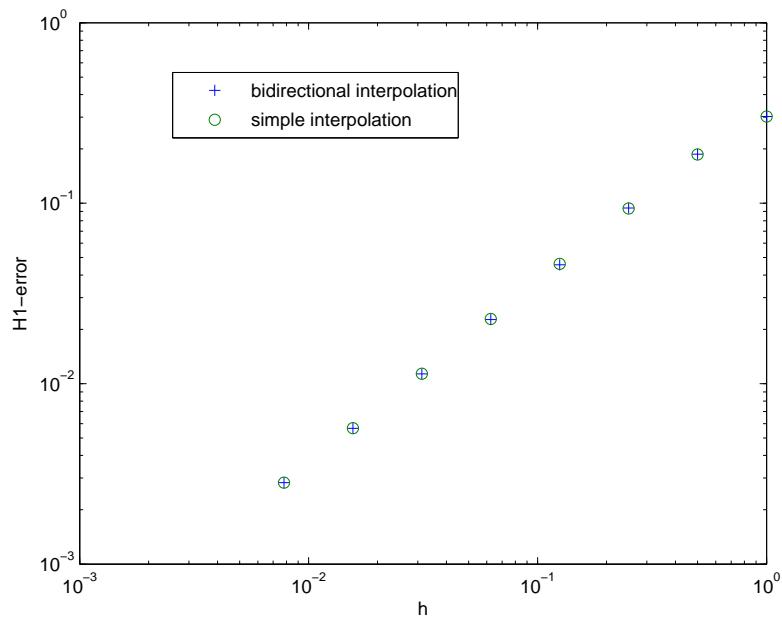


Figure 67: Convergence rate for bidirectional and simple interpolation in the H1-error norm

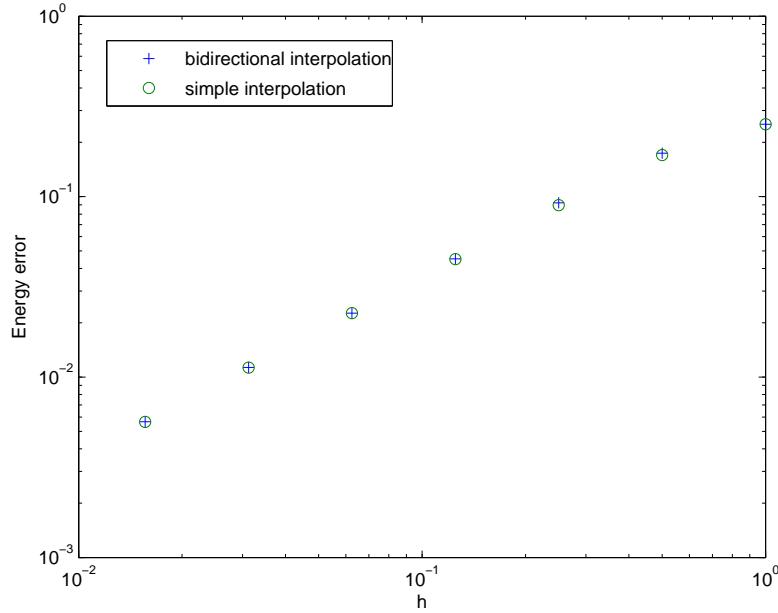


Figure 68: Convergence rate for bidirectional and simple interpolation in the energy error norm

In the H0-error norm, the bidirectional interpolation allows a gain in accuracy. Both methods exhibit the same convergence rate and slope for the H1-error norm, but approximately for the energy error norm.

For the energy error norm, the following inequality holds:

$$\|\mathbf{u} - \mathbf{u}^h\|_E \leq ch^{q-p+1}, \quad (200)$$

where u is the exact solution, \mathbf{u}^h is the finite element solution, c is a problem dependent constant, h is the element size, q is the element order and p is the order of the highest derivative in the weak form.

Readily, the inequality given in (200) does not involve information about the technique used to enforce boundary conditions. Therefore, it makes sense that the simple and bidirectional interpolations lead to the same convergence behavior in the H1- and Energy-norms.

9 Structured vs. Unstructured Meshing

Another interesting study is to investigate the solution behavior of the new algorithm compared to a solution exhibited by a finite element solution using an unstructured mesh, i.e. a mesh that is conform to the boundary.

The test problem of the circular physical domain (Section 7.1) is used here.

9.1 Unstructured Mesh

The circle is spatially discretized using an initial number of 24 elements.

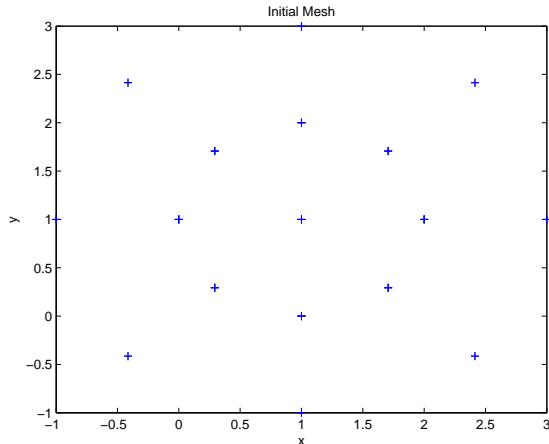


Figure 69: Starting Discretization of the Circle

Refinement happens in levels where each existing element is divided into 4 elements. The nodes situated on the outer edges are forced to lie on the circle.

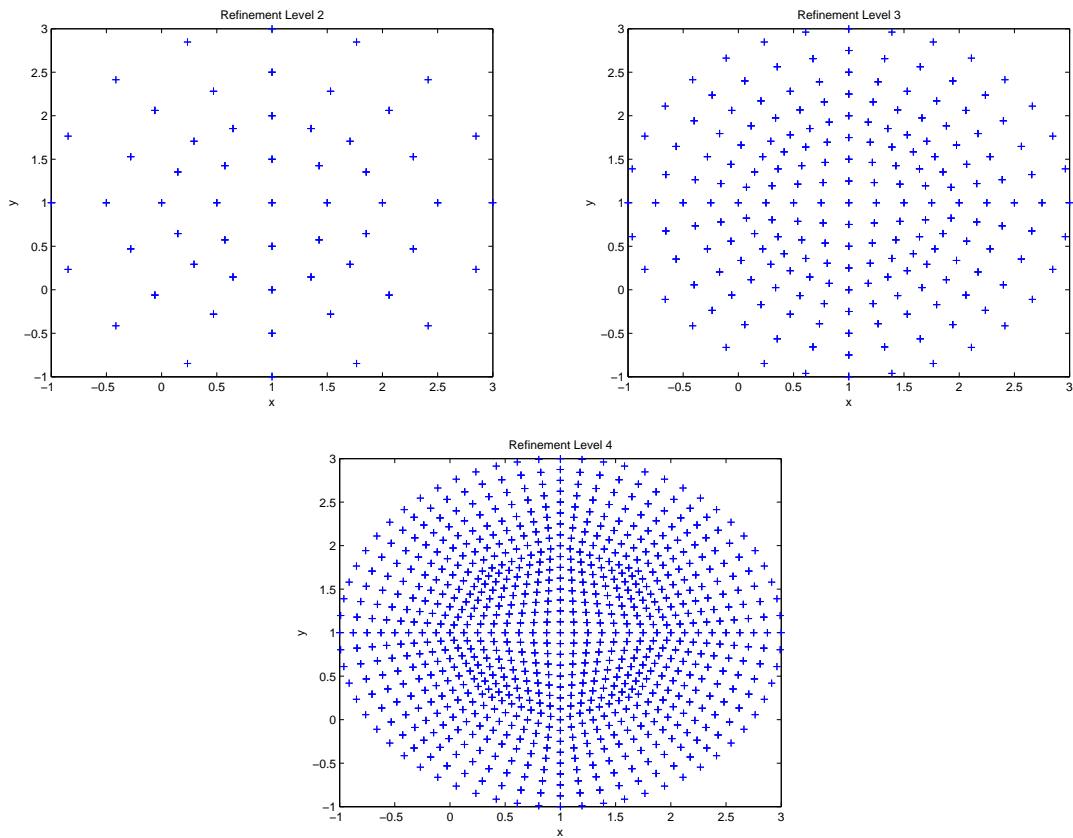


Figure 70: Sequence of Mesh Refinements of the Circle

As the mesh becomes finer, the quality of the elements situated toward the center of the circle becomes bad. Some angles become wider. In order to fairly compare with the partial cells solution, the unstructured mesh quality has to be improved using the techniques presented in sections 9.2 and 9.3.

9.2 Improving Mesh Quality: Neighbor Averaging

An easy technique to improve the mesh quality of unstructured meshes is to recalculate the location of the internal nodes by averaging the location of the neighbor nodes which are connected to it by mesh segments (edges). Averaging is repeated until the location of the internal nodes does not change, or the mesh quality becomes relatively satisfactory.

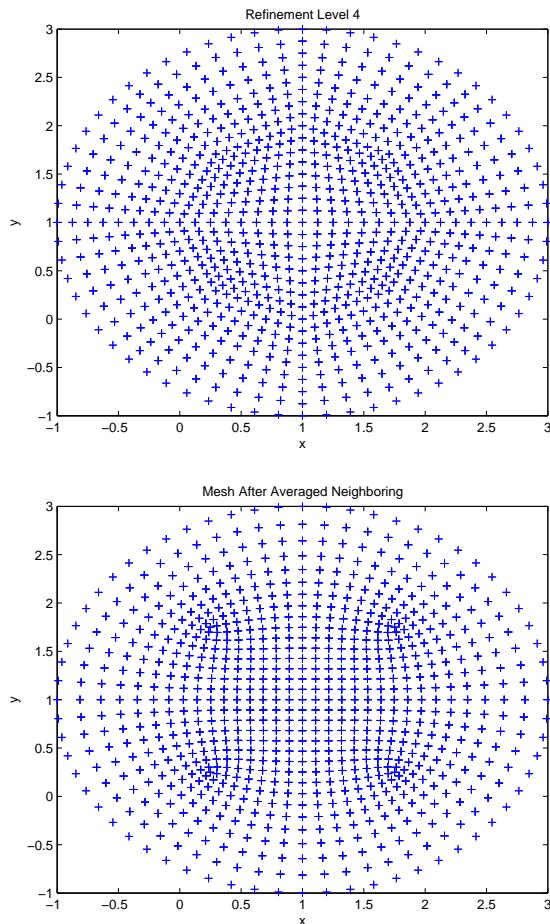


Figure 71: Meshed Circle Before and After Neighbor Averaging (40 times)

As shown in figure 71, the quality of the internal elements is better. The angles are near to be right. However, the elements located at the corner edges could not be improved.

9.3 Improving Mesh Quality: Poisson Smoothing

The Poisson smoothing technique requires the finite difference approximation of partial derivatives that lead to the construction of a system of equations which solves for the “better” position of the nodes. The implementation for the Poisson smoothing method requires a specific type of data structure for storing the nodes and the corresponding neighbors. The currently in the code implemented data structure is not compatible and has to be changed or converted to fit into the Poisson smoothing scheme. This is to be done later.

9.4 Comparison of Convergence Rates: Original Mesh

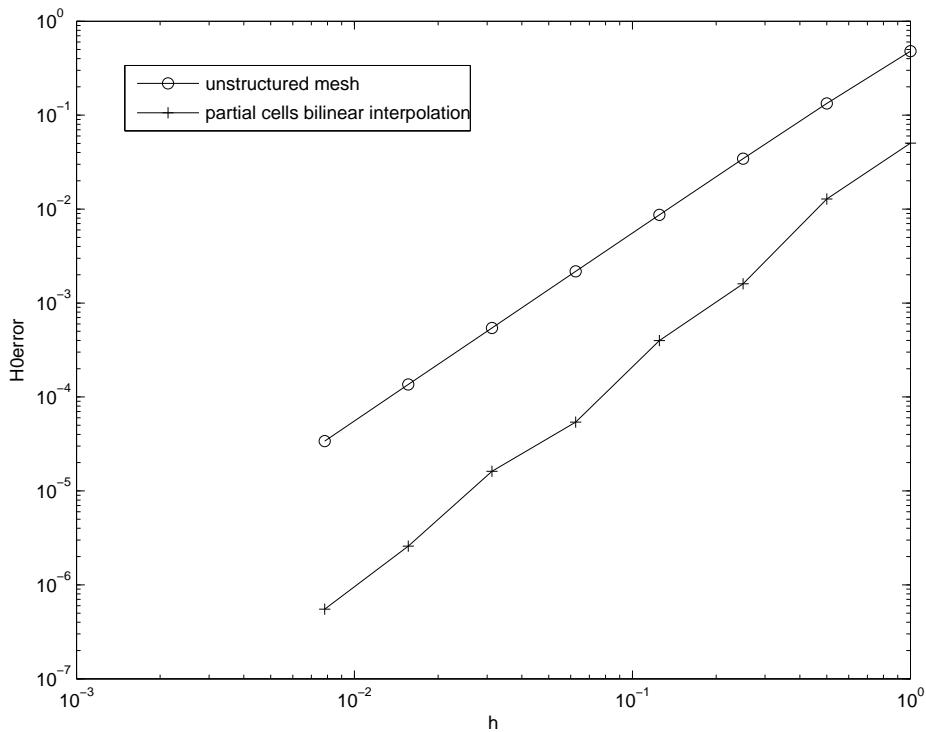


Figure 72: Convergence of Structured vs. Unstructured Mesh in the H0-Error Norm

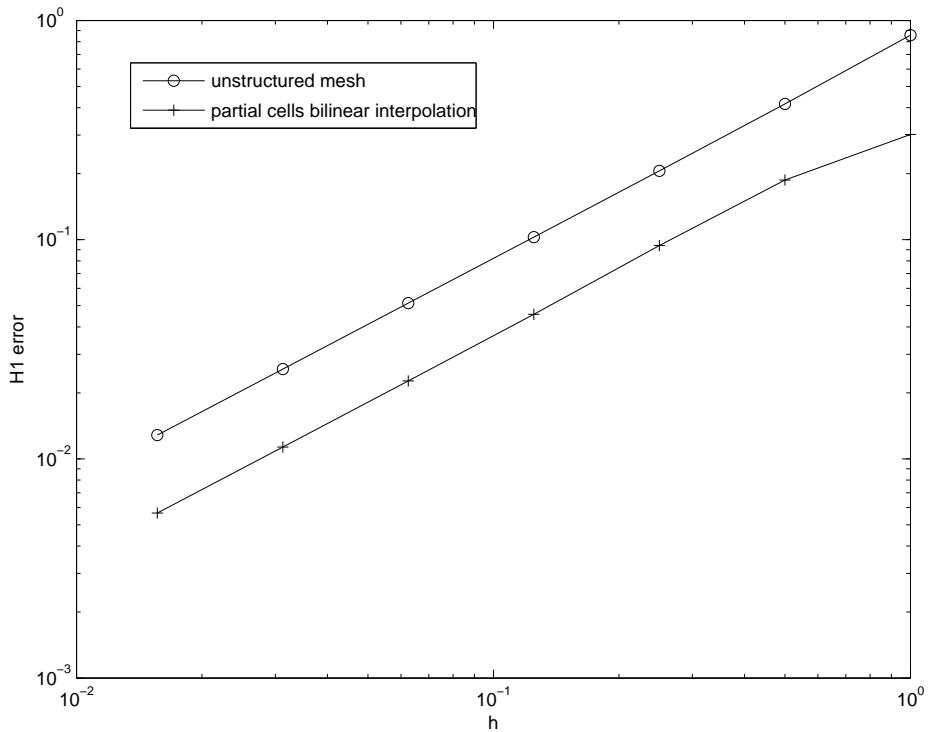


Figure 73: Convergence of Structured vs. Unstructured Mesh in the H1-Error Norm

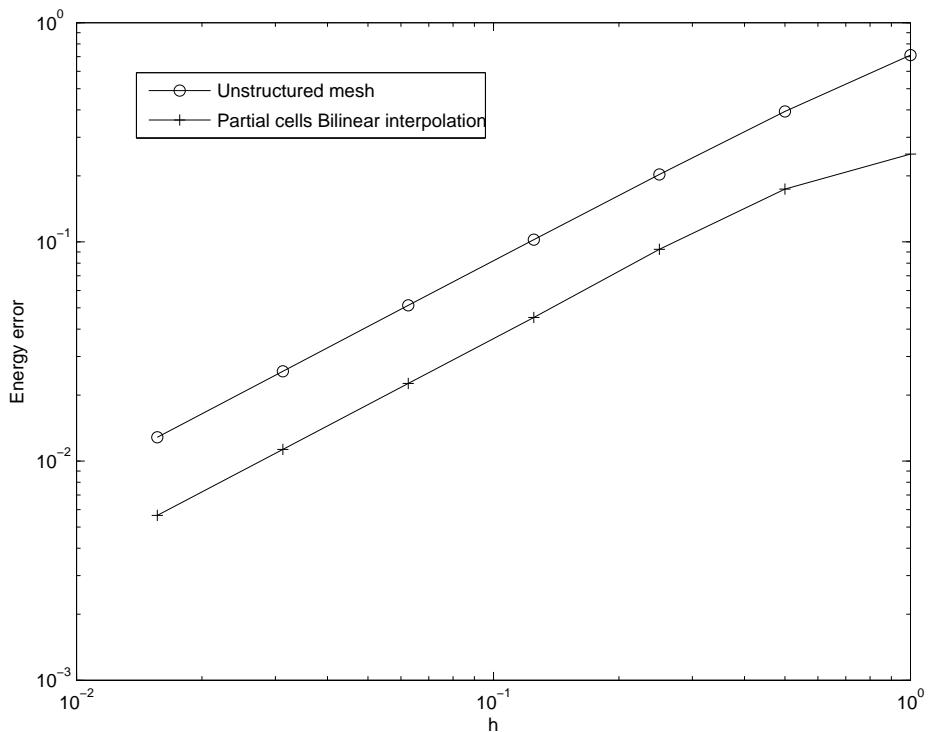


Figure 74: Convergence of Structured vs. Unstructured Mesh in the Energy-Error Norm

Figures 72, 73 and 74 reveal that the partial cells algorithm on a structured grid leads to the same convergence rate as the usual finite element solution on an unstructured mesh. The partial cells algorithm's performance is actually even better. The total error made by the partial cells method is smaller than the error made by the unstructured mesh solution. Interpreting this result regarding the inequality given in equation (200), each of the partial cells solution and the unstructured mesh solution have a different constant c . One should not ignore the effect of the averaged mesh size h on this finding.

9.5 Comparison of Convergence Rates: Neighbor Averaged Mesh

The convergence behavior of the finite element solution using the initial mesh compared to the use of an improved mesh by the neighbor averaging technique reveals no big of a difference at coarser meshes. A tiny difference becomes apparent if the mesh size is very small. Convergence rate in the corresponding error norm are consistent to previous findings.

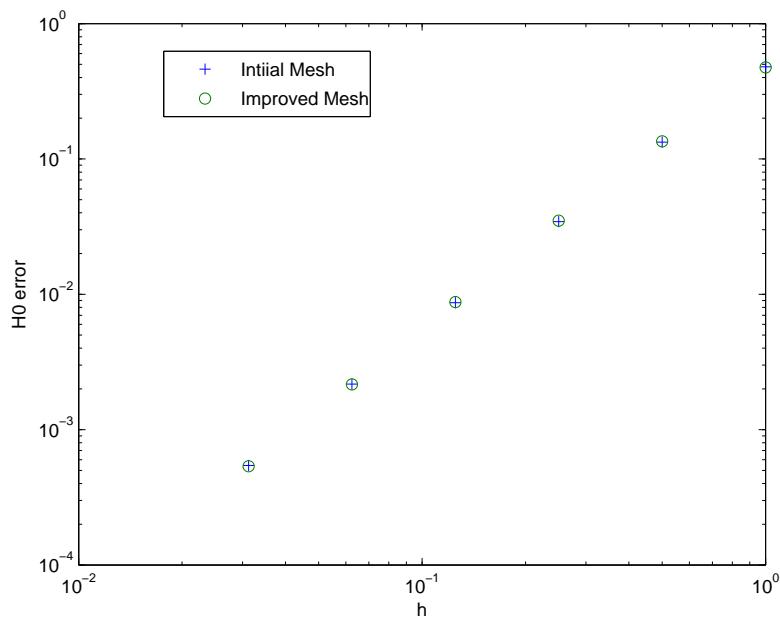


Figure 75: Convergence of Initial vs. Improved Mesh in the H_0 -Error Norm

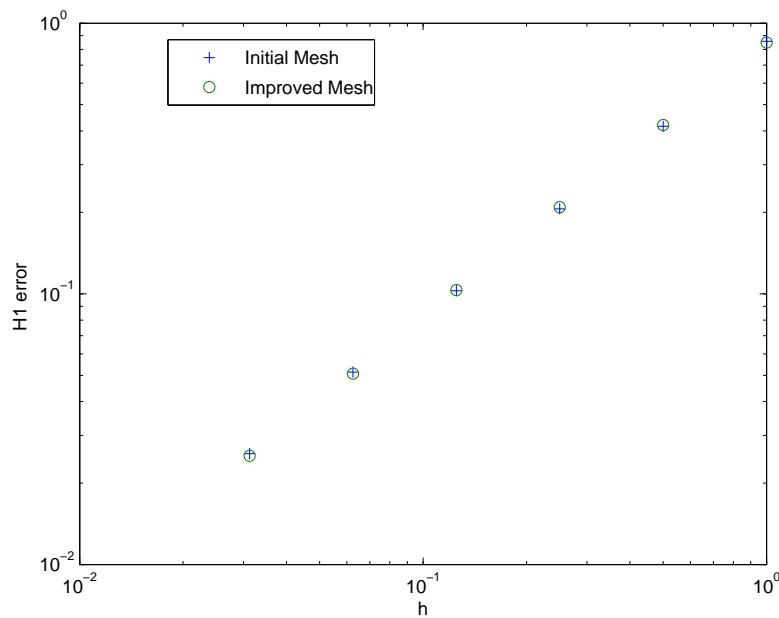


Figure 76: Convergence of Initial vs. Improved Mesh in the H1-Error Norm

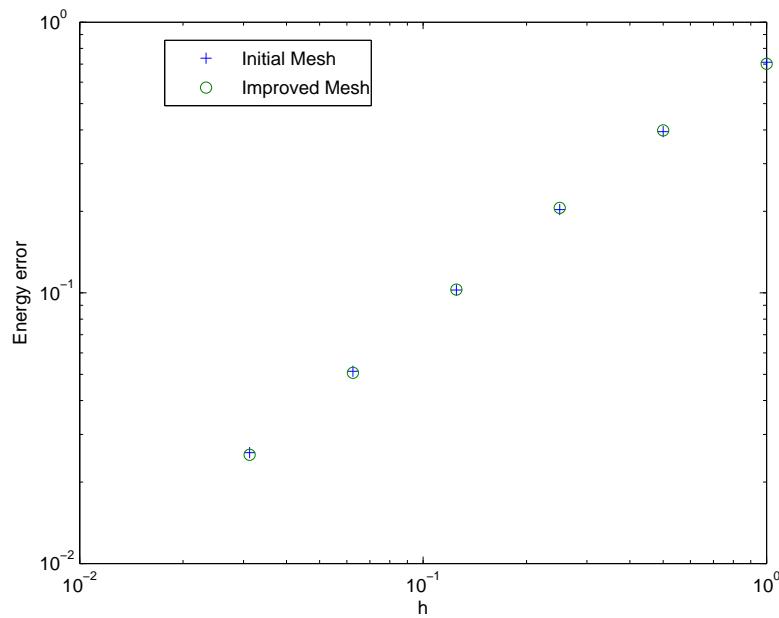


Figure 77: Convergence of Initial vs. Improved Mesh in the Energy Error Norm

10 Torus: Homogeneous Dirichlet Boundary Conditions

We might get the argument, that developing the algorithm for a simple circle and for one single type of boundary conditions, Dirichlet boundary conditions, would not lead to general conclusions.

Therefore, we want to expand the use of this algorithm to different geometries and to different boundary conditions, for which we can derive analytical solutions and use as a verification tool.

In order to prepare for the next section, we will solve the Laplacian on a torus shaped geometry (Figure 78):

$$\nabla \cdot (\nabla u) = 0. \quad (201)$$

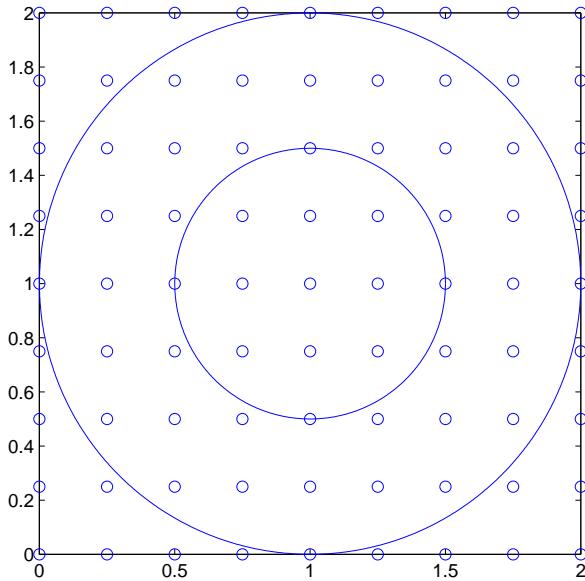


Figure 78: Torus Geometry

The torus is given by the coordinates of its center x_C, y_C , the outer radius R_1 and the inner radius R_2 .

Each of the boundaries is subject to Dirichlet boundary conditions, let's say \bar{u}_1 on the outer boundary and \bar{u}_2 on the inner boundary. We do exclude the inclusion of source terms, since analytical solution are provided in this case for this specific geometry.

10.1 Analytical Solution

The problem can be written in polar coordinates as follows:

$$\nabla \cdot (\nabla u) = 0, \quad (202)$$

$$\nabla \cdot \left(\frac{\partial u}{\partial r}, \frac{1}{r} \frac{\partial u}{\partial \theta} \right) = 0, \quad (203)$$

$$\frac{\partial}{\partial r} \left(\frac{\partial u}{\partial r} \right) + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r} \frac{\partial}{\partial \theta} \left(\frac{1}{r} \frac{\partial u}{\partial \theta} \right) = 0, \quad (204)$$

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \underbrace{\frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2}}_{=0} = 0, \quad (205)$$

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} = 0, \quad (206)$$

whereas it is readily to be concluded that the solution does only vary in the radial direction, and not with the angle. Each radial segment can be seen as a one-dimensional problem, for which the solution vary along the length. Due to symmetry of the problem and of the geometry, each of the segments delivers the same answer. Therefore, one concludes that solution should not depend on the angle θ .

Let $v = \frac{\partial u}{\partial r}$, therefore the problem transforms to be in the independent variable v :

$$\frac{\partial v}{\partial r} + \frac{1}{r} v = 0, \quad (207)$$

$$\frac{\partial v}{\partial r} = -\frac{1}{r} v, \quad (208)$$

$$\frac{\partial v}{v} = -\frac{\partial r}{r}, \quad (209)$$

$$\int \frac{\partial v}{v} = \int -\frac{\partial r}{r} + \tilde{c}, \quad (210)$$

$$\ln(v) = -\ln(r) + \tilde{c}, \quad (211)$$

$$\ln(v) = \ln\left(\frac{1}{r}\right) + \tilde{c}, \quad (212)$$

$$e^{\ln(v)} = e^{\ln\left(\frac{1}{r}\right) + \tilde{c}}, \quad (213)$$

$$e^{\ln(v)} = e^{\ln\left(\frac{1}{r}\right)} e^{\tilde{c}}, \quad (214)$$

$$v = c_1 \frac{1}{r}, \quad (215)$$

$$\frac{du}{dr} = c_1 \frac{1}{r}, \quad (216)$$

$$\int dr = \int c_1 \frac{1}{r} dr, \quad (217)$$

$$u = c_1 \ln(r) + c_2. \quad (218)$$

The constants c_1 and c_2 can be determined using the boundary conditions, i.e.

$$\bar{u}_1 = c_1 \ln(R_1) + c_2, \quad (219)$$

$$\bar{u}_2 = c_1 \ln(R_2) + c_2, \quad (220)$$

$$c_1 = \frac{\bar{u}_1 - \bar{u}_2}{\ln(R_1) - \ln(R_2)}, \quad (221)$$

$$c_2 = \bar{u}_1 - c_1 \ln(R_1). \quad (222)$$

10.2 Any Modifications to the Algorithm Necessary?

The only difference to the problem solved in 7.1 is that there are two boundaries to be taken care of. The ideas presented in section 7.6 can be used as is. One must be careful about which boundary the element is intersection to assign the correct boundary condition. Otherwise, every other step remains intact. This is a first example to show the flexibility and the generality of the algorithm.

10.3 Results

One wants to compare the numerical results with the analytical solution, and investigate the accuracy with different mesh sizes in different error norms (H_0 , H_1 and Energy-Norm). For a concrete convergence study, the following values have been used:

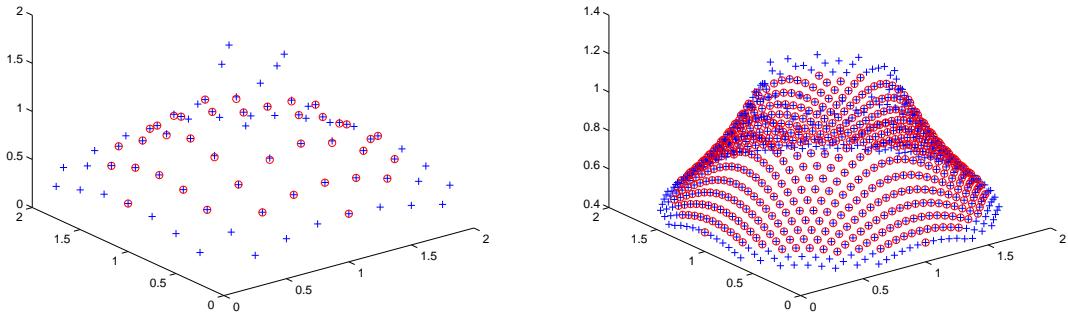
$$R_1 = 1.0, \quad (223)$$

$$R_2 = 0.5, \quad (224)$$

$$\bar{u}_1 = 0.5, \quad (225)$$

$$\bar{u}_2 = 1.0. \quad (226)$$

The numerical solution fits the analytical solution even for coarser meshes. Figure 79 shows both solutions.



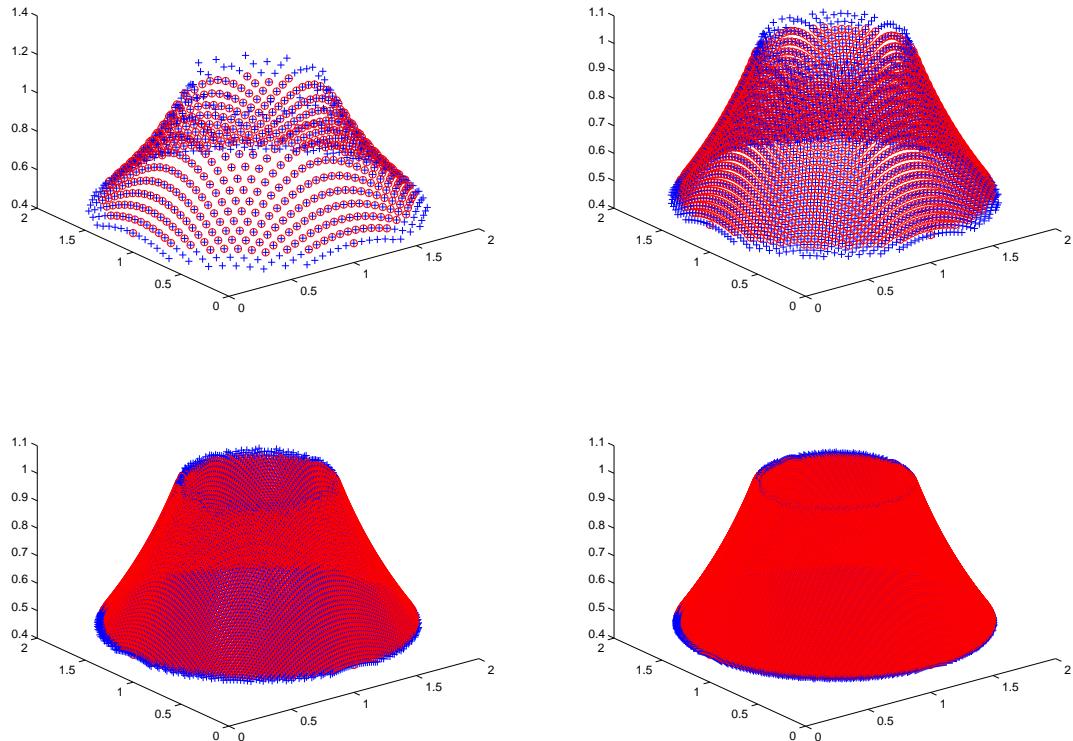


Figure 79: Numerical(+) vs. Analytical (o) Solution

10.3.1 Convergence in H_0 -norm

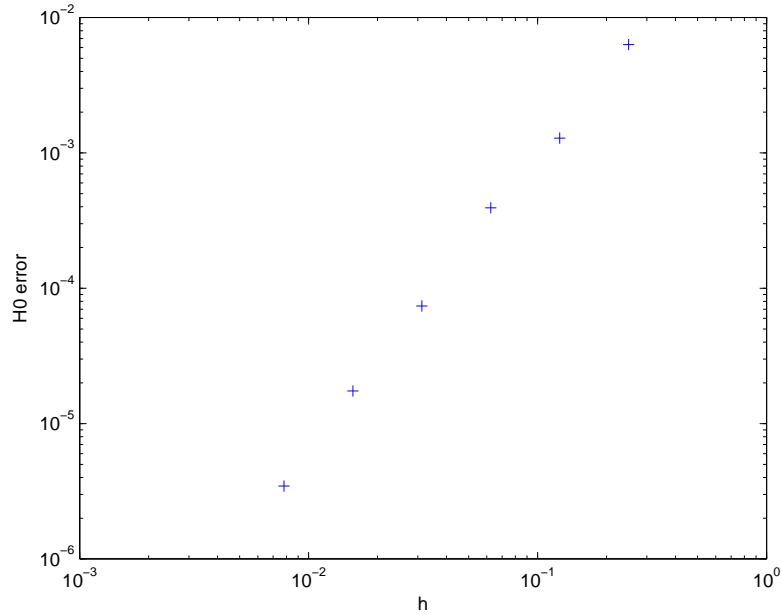


Figure 80: Convergence in the H_0 -Norm

The convergence in the H_0 -norm is quadratic, and consistent with earlier findings.

10.3.2 Convergence in H_1 -norm

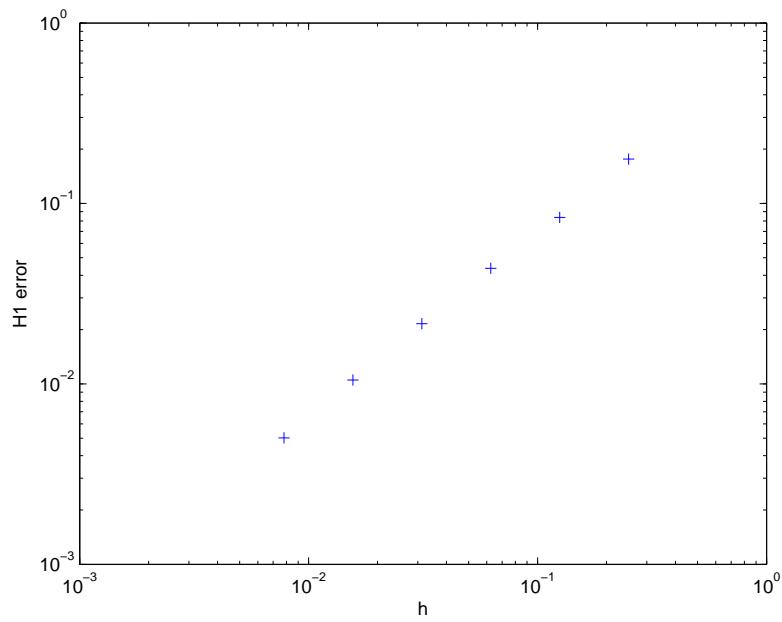


Figure 81: Convergence in the H_1 -Norm

The convergence in the H_1 -norm is linear, and consistent with earlier findings.

10.3.3 Convergence in Energy-Norm

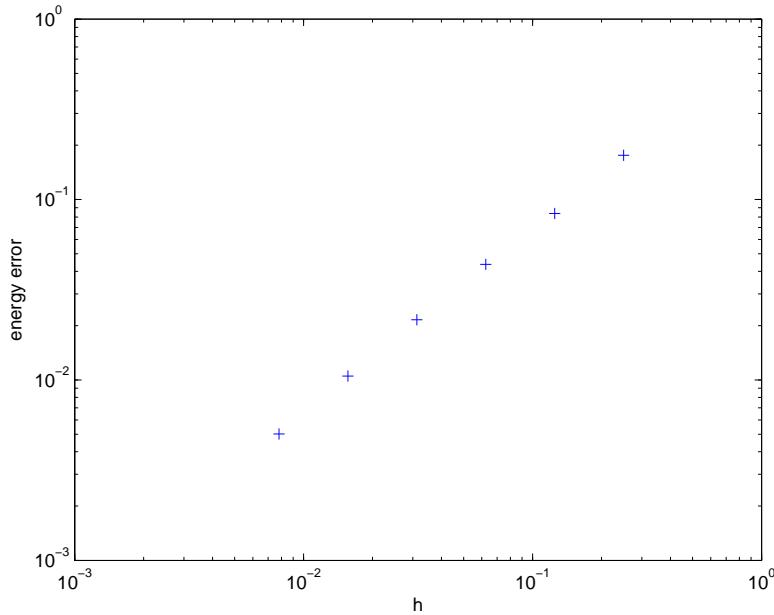


Figure 82: Convergence in the Energy-Norm

The convergence in the energy-norm is linear, and consistent with earlier findings.

11 Torus: Homogeneous Neumann Boundary Conditions

The example problem treated in section 10 is now slightly modified, such that the inner boundary of the torus is now subject to Neumann boundary conditions:

$$\frac{\partial u}{\partial n}|_{R_2} = \nabla u \cdot \mathbf{n}|_{R_2} = \bar{q}. \quad (227)$$

The only modification to the algorithm is to modify the interpolation equation that imposes the boundary condition from a linear collocation technique to a gradient relationship that accounts for the normal direction of the boundary that is exposed to the Neumann boundary conditions. Let's say, we are given the following interpolation equation to impose the value

\bar{u} at the boundary:

$$\bar{u} = \sum_{i=1}^4 \omega_i(x, y) u_i, \quad (228)$$

$$\omega_1(x, y) = \frac{(y_2 - y)}{(y_2 - y_1)} \frac{(x_2 - x)}{(x_2 - x_1)}, \quad (229)$$

$$\omega_2(x, y) = \frac{(y_2 - y)}{(y_2 - y_1)} \frac{(x - x_1)}{(x_2 - x_1)}, \quad (230)$$

$$\omega_3(x, y) = \frac{(y - y_1)}{(y_2 - y_1)} \frac{(x_2 - x)}{(x_2 - x_1)}, \quad (231)$$

$$\omega_4(x, y) = \frac{(y - y_1)}{(y_2 - y_1)} \frac{(x - x_1)}{(x_2 - x_1)}, \quad (232)$$

$$(233)$$

where $\omega(x, y)$ are weights coming from the linear interpolation relationship and do depend on the spatial location (x, y) of interest. u_i are the nodal solutions. x_1, x_2, y_1 and y_2 are the minimal and maximal spatial locations of the grid cell of interest.

The boundary condition given in (227) can be imposed using the cartesian coordinate system as follows:

$$\nabla u \cdot \mathbf{n}|_{R_2} = \left(\begin{array}{c} \sum_{i=1}^4 \omega_{i,x}(x, y) u_i \\ \sum_{i=1}^4 \omega_{i,y}(x, y) u_i \end{array} \right) \cdot \mathbf{n}|_{R_2}, \quad (234)$$

$$= \left(\begin{array}{c} \sum_{i=1}^4 \omega_{i,x}(x, y) u_i \\ \sum_{i=1}^4 \omega_{i,y}(x, y) u_i \end{array} \right) \cdot \left(\begin{array}{c} n_x \\ n_y \end{array} \right) |_{R_2}, \quad (235)$$

$$= (\sum_{i=1}^4 \omega_{i,x}(x, y) u_i) \tilde{n}_x + (\sum_{i=1}^4 \omega_{i,y}(x, y) u_i) \tilde{n}_y, \quad (236)$$

$$= \sum_{i=1}^4 (\omega_{i,x}(x, y) \tilde{n}_x + \omega_{i,y}(x, y) \tilde{n}_y) u_i, \quad (237)$$

$$= \sum_{i=1}^4 \alpha_i u_i. \quad (238)$$

$$\omega_{1,x} = \frac{-(y_2 - y)}{(y_2 - y_1)(x_2 - x_1)}, \quad \omega_{1,y} = \frac{-(x_2 - x)}{(y_2 - y_1)(x_2 - x_1)}, \quad (239)$$

$$\omega_{2,x} = \frac{(y_2 - y)}{(y_2 - y_1)(x_2 - x_1)}, \quad \omega_{2,y} = \frac{-(x - x_1)}{(y_2 - y_1)(x_2 - x_1)}, \quad (240)$$

$$\omega_{3,x} = \frac{-(y - y_1)}{(y_2 - y_1)(x_2 - x_1)}, \quad \omega_{3,y} = \frac{(x_2 - x)}{(y_2 - y_1)(x_2 - x_1)}, \quad (241)$$

$$\omega_{4,x} = \frac{(y - y_1)}{(y_2 - y_1)(x_2 - x_1)}, \quad \omega_{4,y} = \frac{(x - x_1)}{(y_2 - y_1)(x_2 - x_1)} \quad (242)$$

11.0.4 Analytical Solution

We want to impose:

$$u|_{R_1} = \bar{u}, \quad (243)$$

$$\nabla u \cdot \mathbf{n}|_{R_2} = \bar{q} = \left(\frac{\frac{\partial u}{\partial r}}{\frac{1}{r} \frac{\partial u}{\partial \theta}} \right) \cdot (-\mathbf{e}_r)|_{R_2} = -\frac{\partial u}{\partial r}|_{R_2}, \quad (244)$$

to:

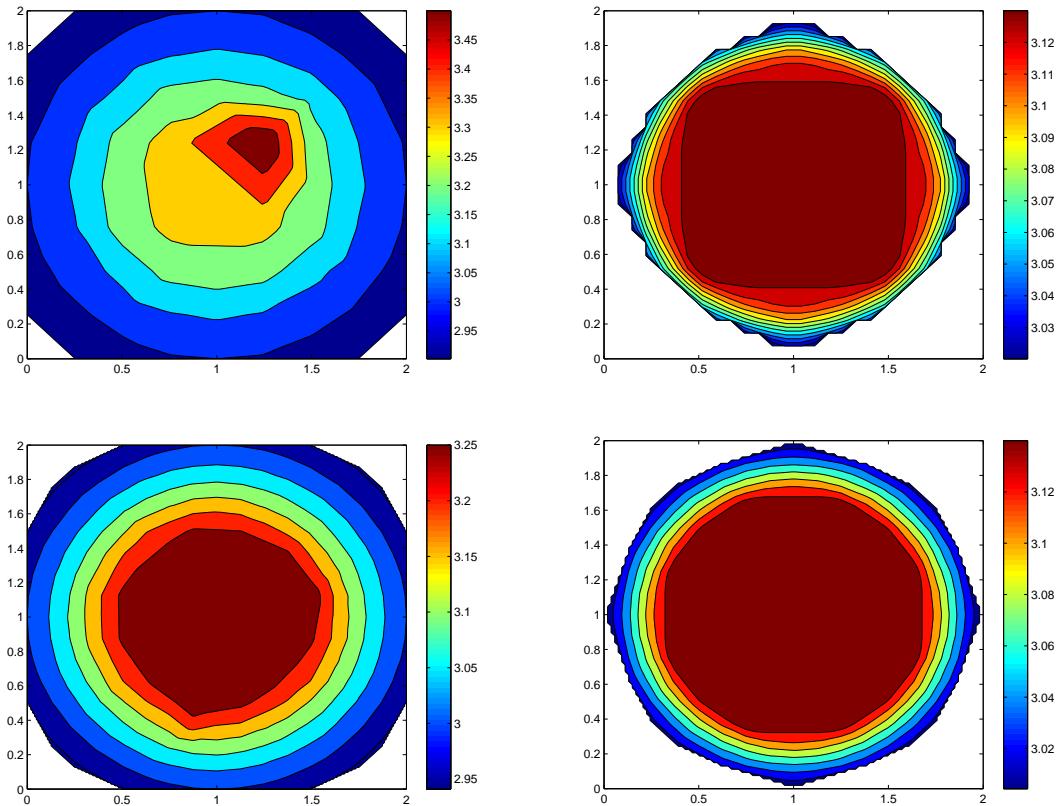
$$u = c_1 \ln(r) + c_2, \quad (245)$$

$$= (-R_2 \bar{q})(\ln(r) - \ln(R_1)) + \bar{u}. \quad (246)$$

11.1 Debugging

The first implementation of the torus under mixed boundary conditions did not lead to a consistent convergence rate compared to the analytical solution. Contour plots of the numerical and the exact solution for the same spatial discretization are given for the specific problem:

$$R_1 = 1.0, \quad R_2 = \frac{2}{3}, \quad \bar{u} = 3.0, \quad \bar{q} = 0.6, \quad (247)$$



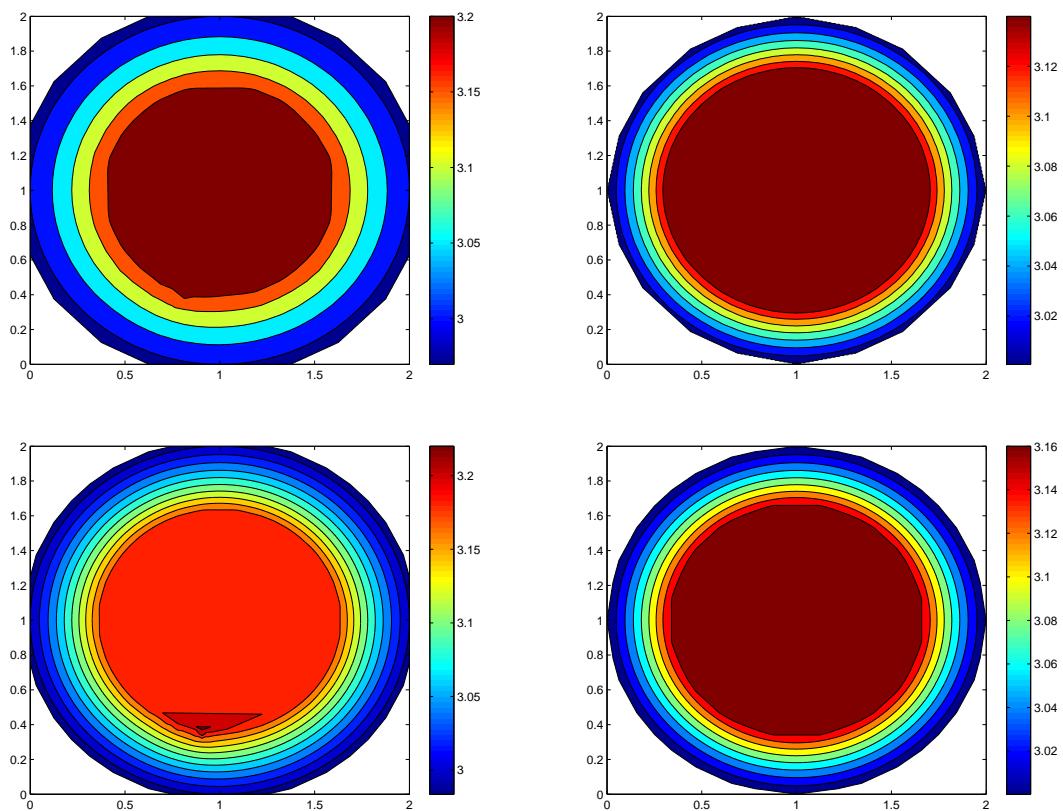


Figure 83: Contour Plots of the Numerical Solution (left) vs. the Analytical Solution (right) for the Same Spatial Resolution. The Grid Resolutions are 8x8, 16x16, 32x32, 64x64.

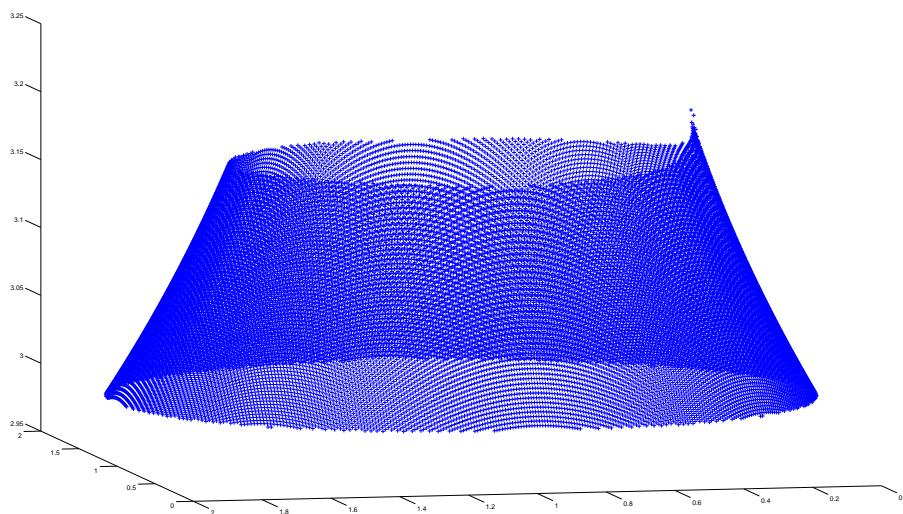
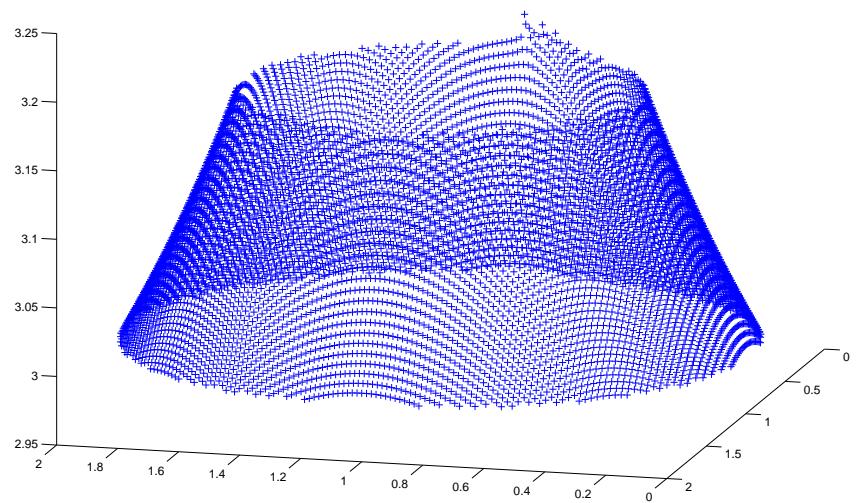


Figure 84: Numerical Solution for the Spatial Resolutions 128x128 and 256x256

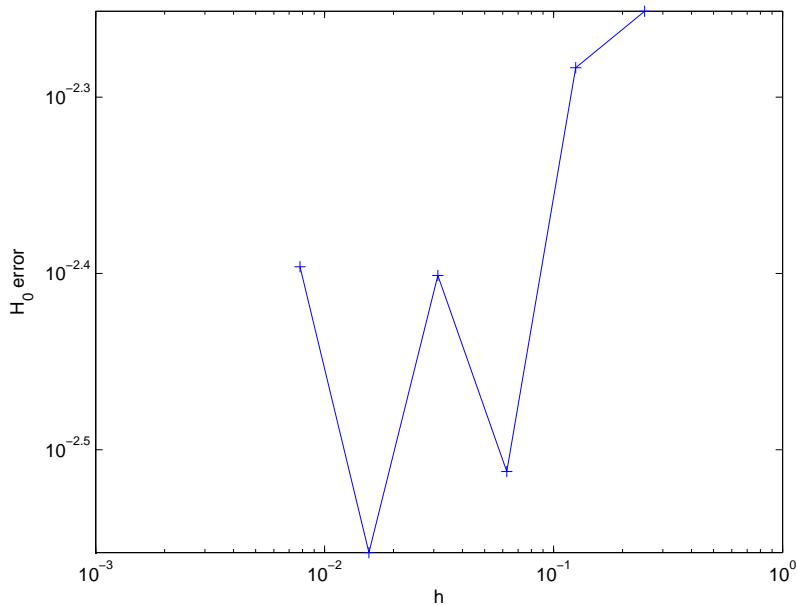


Figure 85: Divergence of the Numerical Solution

11.1.1 Suspection

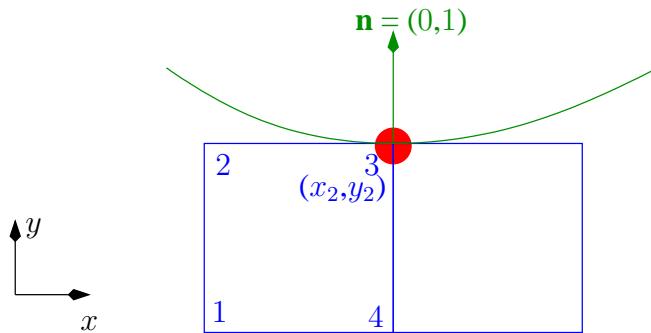


Figure 86: Critical Case

In the case shown in figure 86, the hanging node is an intersection point itself. This was a convenient case for imposing Dirichlet boundary conditions. The value of the Dirichlet boundary condition is directly assigned to the node.

In this case, we have:

$$\omega_{1,x} = 0, \quad \omega_{1,y} = 0, \quad (248)$$

$$\omega_{2,x} = 0, \quad \omega_{2,y} \neq 0, \quad (249)$$

$$\omega_{3,x} \neq 0, \quad \omega_{3,y} = 0, \quad (250)$$

$$\omega_{4,x} \neq 0, \quad \omega_{4,y} \neq 0, \quad (251)$$

$$(252)$$

Thus,

$$\alpha_1 = 0, \quad \alpha_2 \neq 0, \quad \alpha_3 = 0, \quad \alpha_4 \neq 0 \quad (253)$$

Different ideas for debugging the code are investigated.

11.1.2 Zero Neumann Boundary Conditions

Since the Dirichlet-Dirichlet boundary value problem on a torus has worked bug-free, the only error source should be in the method of imposing the Neumann boundary conditions. In order to avoid the effect of such a modification, the Neumann boundary condition is set to a value of $\bar{q} = 0$.

The analytical solution reads then:

$$u = \bar{u}. \quad (254)$$

The algorithm exactly the analytical solution. Therefore, the error must be, but not exclusively, in the collocation condition of the Neumann boundary condition.

11.1.3 Flipping the Neumann and the Dirichlet Boundary Sides

We want to impose:

$$\nabla u \cdot \mathbf{n}|_{R_1} = \bar{q} = \left(\begin{array}{c} \frac{\partial u}{\partial r} \\ \frac{1}{r} \frac{\partial u}{\partial \theta} \end{array} \right) \cdot (\mathbf{e}_r)|_{R_1} = \frac{\partial u}{\partial r}|_{R_1}, \quad (255)$$

$$u|_{R_2} = \bar{u}, \quad (256)$$

to:

$$u = c_1 \ln(r) + c_2, \quad (257)$$

$$= (R_1 \bar{q})(\ln(r) - \ln(R_2)) + \bar{u}. \quad (258)$$

The spatial discretization for the finite element solution is fine enough such that a grid cell is exclusively intersected by a single boundary domain, i.e. associated with a single boundary condition. Therefore, flipping the nature of the boundary conditions between the boundaries did not help identify the nature of the numerical error.

11.1.4 Circle: Homogeneous Neumann Boundary Conditions and Dirichlet at One Single Point

We want to reduce the problem to a single boundary on which Neumann boundary conditions are applied. The geometry becomes now a circle of radius R instead of the donut. We solve:

$$A \nabla \cdot \nabla u + f = 0. \quad (259)$$

Solutions to problems with solely Neumann boundary conditions are non-unique. A Dirichlet boundary condition is imposed on a point in the circular domain, let's say the center of the circle.

The boundary conditions read therefore:

$$A \nabla u \cdot \mathbf{n}|_R = \bar{q}, \quad (260)$$

$$u|_0 = \bar{u}. \quad (261)$$

The analytical solution to this problem

$$u = -\frac{f}{4A}r^2 + C_2 \quad (262)$$

is restrictive, since the choice of the flux boundary condition can not be made independently of the source flux.

$$u|_0 = \bar{u} = C_2, \quad (263)$$

$$A \frac{\partial u}{\partial r}|_R = \bar{q} = -A \frac{f}{2A}R = -\frac{fR}{2}. \quad (264)$$

The latter equation shows that the flux source and the flux boundary conditions can not be chosen independently. Therefore, we restrain from using this sample problem to debug the algorithm on imposing Neumann boundary conditions on partial cells.

11.1.5 Dirichlet-Dirichlet-Problem as Input

The idea behind this test was to solve the Dirichlet-Dirichlet-Problem from above, extract the resulting fluxes from one of the Dirichlet boundaries, impose those fluxes as a Neumann-Dirichlet-Problem. The test should be successful if the resulting displacement at the Neumann boundary from the Neumann-Dirichlet-Problem should compare with the input Dirichlet boundary conditions of the Dirichlet-Dirichlet-Problem.

However, as much as this idea seems to be legitimate, numerical and truncation errors would not lead to a fair comparison.

11.2 Working Version

The Neumann-Dirichlet problem on a Torus geometry has been fixed. Changes to the algorithm were necessary. The suspected problem in section 11.1.1 was effectively a cause for

head-ache. The type of elements where all nodes lie in the physical domain and a single node lies on the domain boundary were convenient for the case of imposing Dirichlet boundary conditions. The node lying on the physical boundary was considered a hanging node. And therefore, the element itself was treated as a partial element. Bilinear interpolation equations evaluated at the hanging node led to an exact condition for imposing the Dirichlet boundary condition at the hanging node. However, such kind of elements were problematic for imposing Neumann boundary conditions. At such kind of elements, where the intersection with the boundary is a single point, there is no notion of imposed force since the area associated to the considered element on which the traction is applied is zero (a single point). The interpolation equation does not assign a contribution to the hanging node but to some nodes included in the physical domain. Moreover, the original algorithm did not save the contribution of all elements to the same node. Looping over the elements happened in the same order the elements were stored, and the last element to which the considered hanging node belongs assigned an equation to the degree of freedom. All other contributions from other elements were wiped out. As for the Dirichlet boundary condition, this was not a problem, since the interpolation equation has a local (pointwise) feature, and no element was preferred. The Neumann boundary condition carries information over a certain region. Therefore, it is necessary to carry information the right way and consider the contribution from all sharing elements.

11.2.1 Changes to the Algorithm

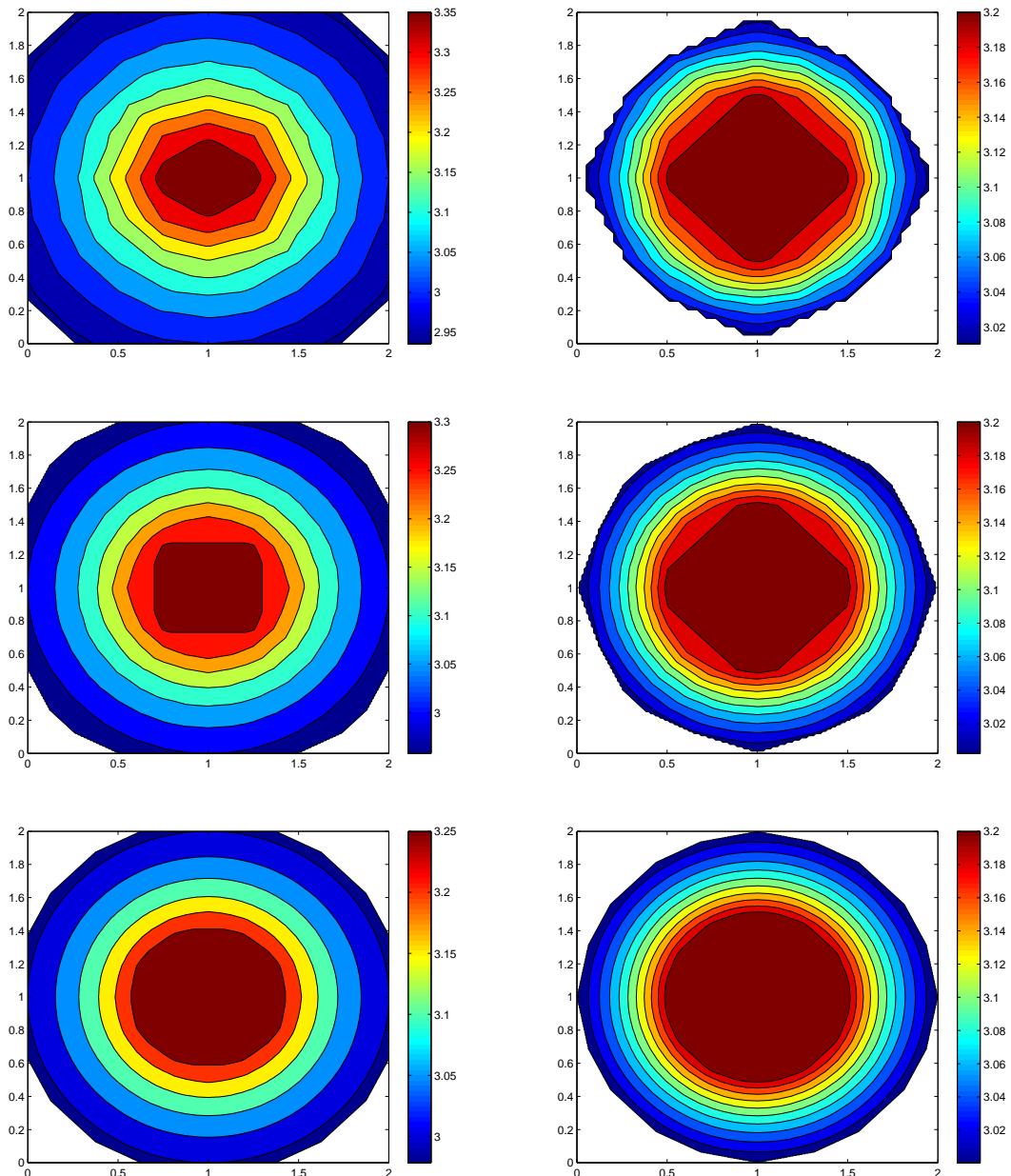
After assembling the material contribution to all fully covered and partially covered elements, the equations associated with hanging nodes have to be wiped out. Differentiation between hanging nodes associated with Dirichlet boundary conditions and Neumann boundary conditions have to be made. As from before, the boundary conditions are imposed through a collocation method rather than through the weak form.

If a node is associated with Dirichlet boundary conditions, exceptional elements having a single node lying on the physical boundary can be treated as a partial element. Interpolation conditions will induce the local feature and can replace the equation associated with the hanging node degree of freedom. All elements that do share the same handing node contribute to the final equation associated with the hanging node degree of freedom. All interpolation equations coming from all contributing elements are summed together. This is in fact an assembly of all interpolation equations.

If a node is associated with a Neumann boundary condition, only elements that do intersect the boundary twice, i.e. elements that do at least have a hanging node is lying exclusively outside of the physical domain, do contribute to the degree of freedom equation. Contributions from appropriate elements are summed to lead to the final degree of freedom equation. In order to facilitate this approach, a new type of elements have to be identified. Until now, we had fully covered element, partial elements and out of the domain elements. Partial elements are divided into two categories, elements were at least a single node is exclusively outside of the physical domain and elements were all nodes lie inside of the physical domain where only a single node lies exactly on the physical boundary.

11.2.2 Results

The numerical solution given by the earlier version of the algorithm did not show symmetry for coarse grids, which was unintuitive. The new algorithm exhibits the symmetry property in the solution even for coarse grids. The contour plots of the numerical solution and the exact solution are given in figure 87:



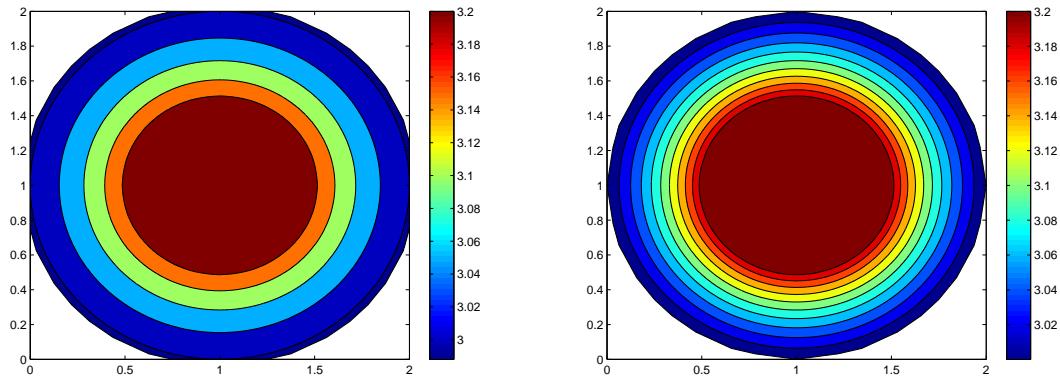


Figure 87: Contour Plots of the Numerical Solution (left) vs. the Analytical Solution (right) for the Same Spatial Resolution. The Grid Resolutions are 8x8, 16x16, 32x32, 64x64.

11.2.3 Convergence Behavior

The convergence behavior of the solution is studied for different grid sizes in the H0- and the H1- error norms (Figures 88 and 89). The convergence is nearly quadratic in the H0-norm and perfectly linear in the H1-norm.

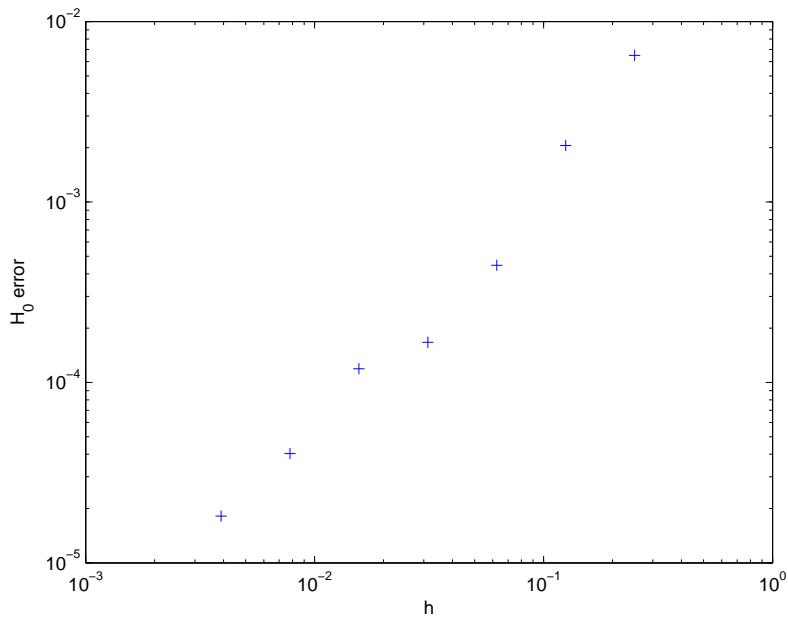


Figure 88: Convergence Behavior in the H0- error norm as a function of the grid cell size.

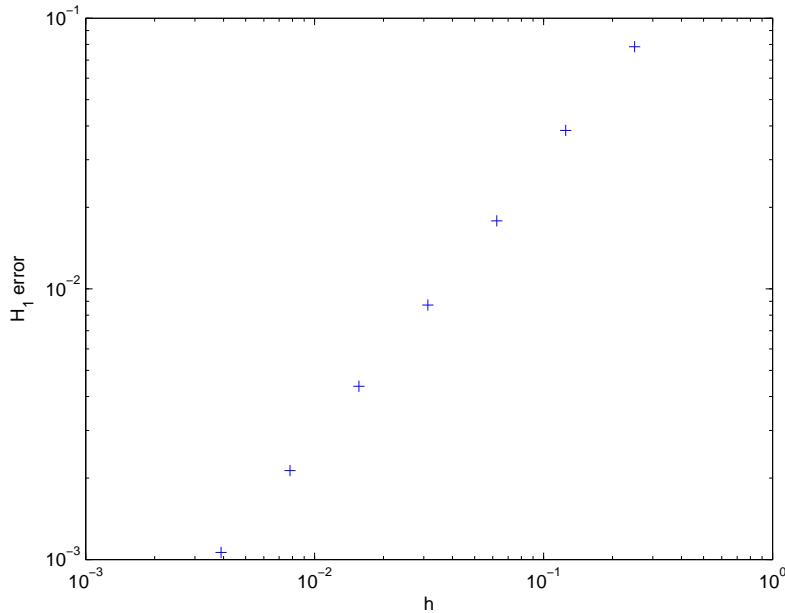


Figure 89: Convergence Behavior in the H1- error norm as a function of the grid cell size.

The convergence study in the H0-norm exhibits a slope of 1.5 with respect to the element size. This might be surprising because of the quadratic convergence order in the H0-norm for problems with Dirichlet boundary conditions.

The order of convergence in the H1-norm is linear, similar to the case of Dirichlet boundary conditions.

We want to investigate more the reason behind the loss of a half of order of convergence in the convergence study with respect to the H0-norm.

Therefore, we want to go back to the one-dimensional case and look at the convergence rate of Neumann problems. After that, we want to solve a Neumann-problem using regular finite elements, i.e. without using the concept of partial cells. For this specific case, we will look at the convergence behavior in the H0-norm.

11.2.4 Dirichlet-Dirichlet-Problem with Finite Volumes and Partial Cells 1d

The problem to be solved is:

$$\frac{\partial^2 u}{\partial x^2} = 0 \quad \text{on} \quad 0 \leq x \leq 1, \quad (265)$$

$$u|_{x=0} = \bar{u}_1 = 0.0, \quad (266)$$

$$u|_{x=1} = \bar{u}_2 = 0.5, \quad (267)$$

$$\alpha = 0.9. \quad (268)$$

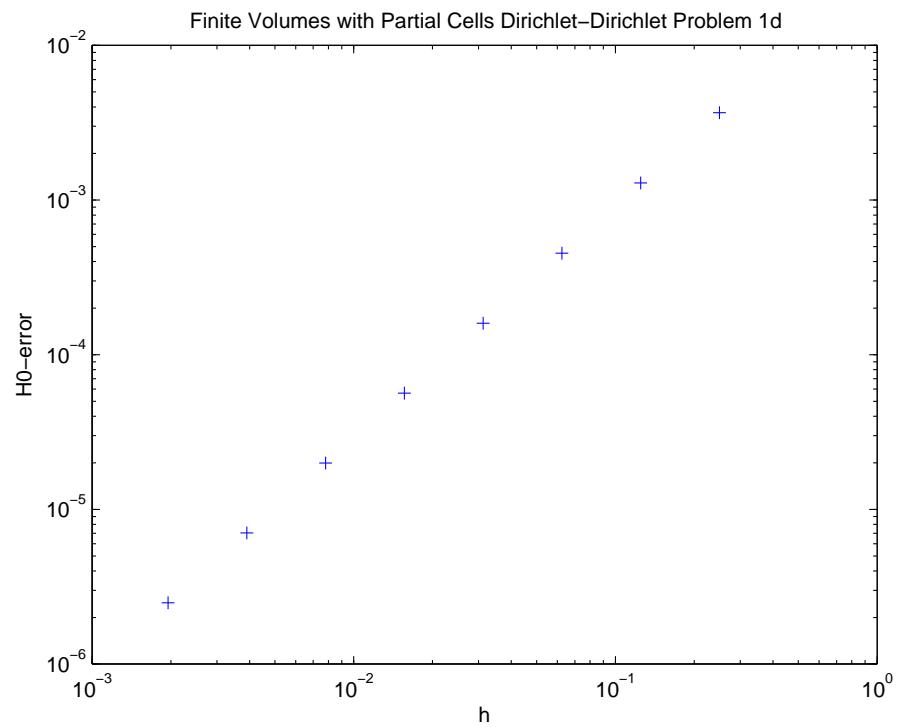


Figure 90: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Volumes and Partial Cells: H0-convergence

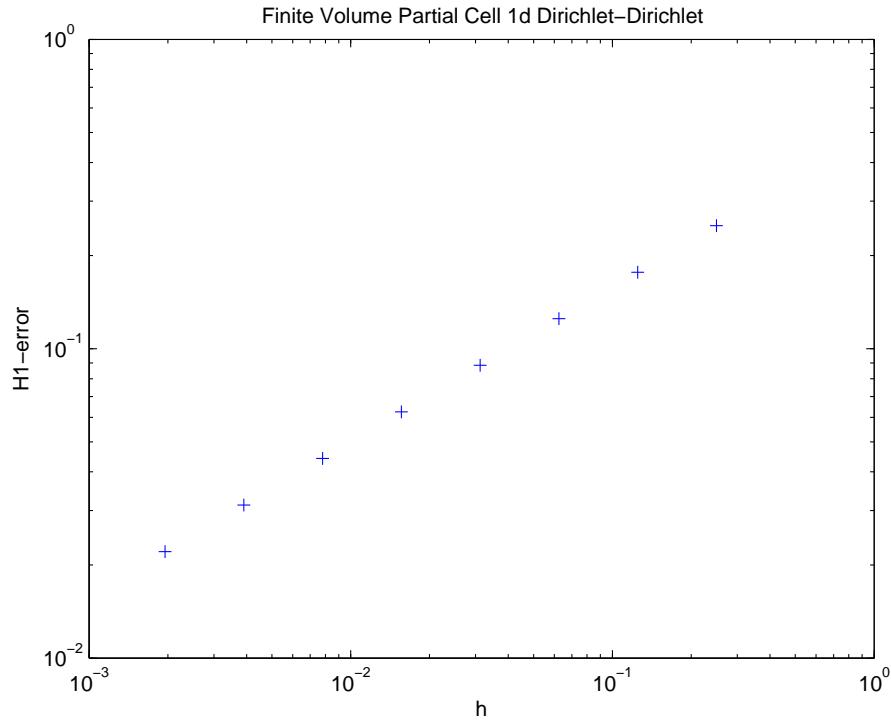


Figure 91: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Volumes and Partial Cells: H1-convergence

The study of this problem confirms the order of convergence in the H0-norm gotten in the two-dimensional cases obtained with the partial cells concept. The slope of the convergence curve in the H0-norm is of the order of 1.498859146854158. We are however not able to compare the convergence in the H1 norm, since the finite volume does not give a good approximation for the slopes of the solution over a control volume.

11.2.5 Dirichlet-Neuman-Problem with Finite Volumes and Partial Cells 1d

The problem to be solved is:

$$\frac{\partial^2 u}{\partial x^2} = -\frac{1}{3} \quad \text{on} \quad 0 \leq x \leq 1, \quad (269)$$

$$u|_{x=0} = \bar{u}_1 = 0.0, \quad (270)$$

$$\frac{\partial u}{\partial x}|_{x=1} = \bar{u}_2 = 0.5, \quad (271)$$

$$\alpha = 0.9. \quad (272)$$

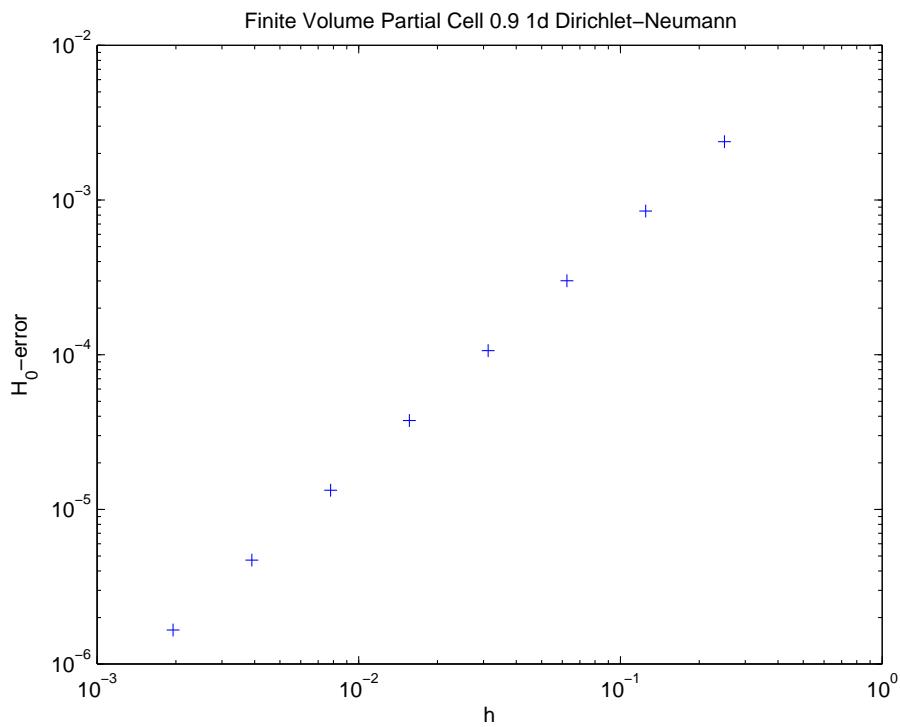


Figure 92: One-dimensional Dirichlet-Neumann-Problem solved with Finite Volumes and Partial Cells: H_0 -convergence

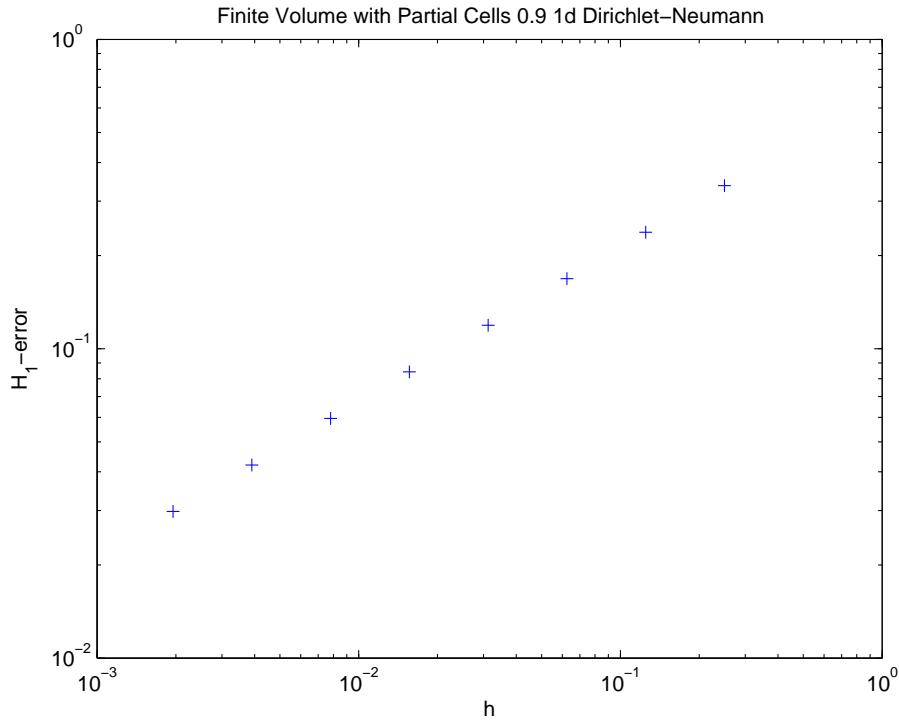


Figure 93: One-dimensional Dirichlet-Neumann-Problem solved with Finite Volumes and Partial Cells: H1-convergence

The slope of the convergence curve in the H0-norm is of the order of 1.498859146854158. The finite volume approach combined with partial cells keeps the same order of convergence independently of the type of boundary conditions imposed. The order of convergence in the H1-norm is unchanged compared to the case of a Dirichlet-Dirichlet-problem. The slope of the convergence curve is of the order of 0.5.

11.2.6 Dirichlet-Dirichlet-Problem with Finite Elements and Partial Cells 1d

The problem to be solved is:

$$\frac{\partial^2 u}{\partial x^2} = -x \quad \text{on } 0 \leq x \leq 1, \quad (273)$$

$$u|_{x=0} = \bar{u}_1 = 0.0, \quad (274)$$

$$u|_{x=1} = \bar{u}_2 = 0.5, \quad (275)$$

$$\alpha = 0.5. \quad (276)$$

The right hand side is taken explicitly to be linear in space, such that the solution would become quadratic. This way, we do not deal with round-off errors since the finite element method with partial cells did hit the exact solution. The overall error was dominated by the round-off errors. The convergence behavior could not be concluded to be consistent for a

certain order. In the quadratic solution case, round off errors are negligible and conclusions can be made consistently.

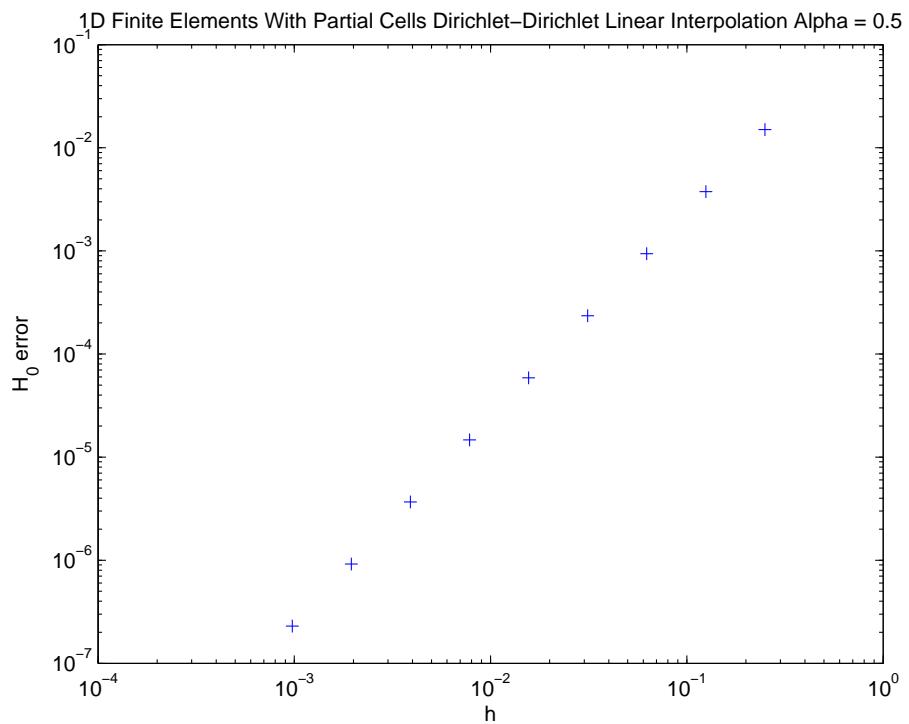


Figure 94: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H₀-convergence

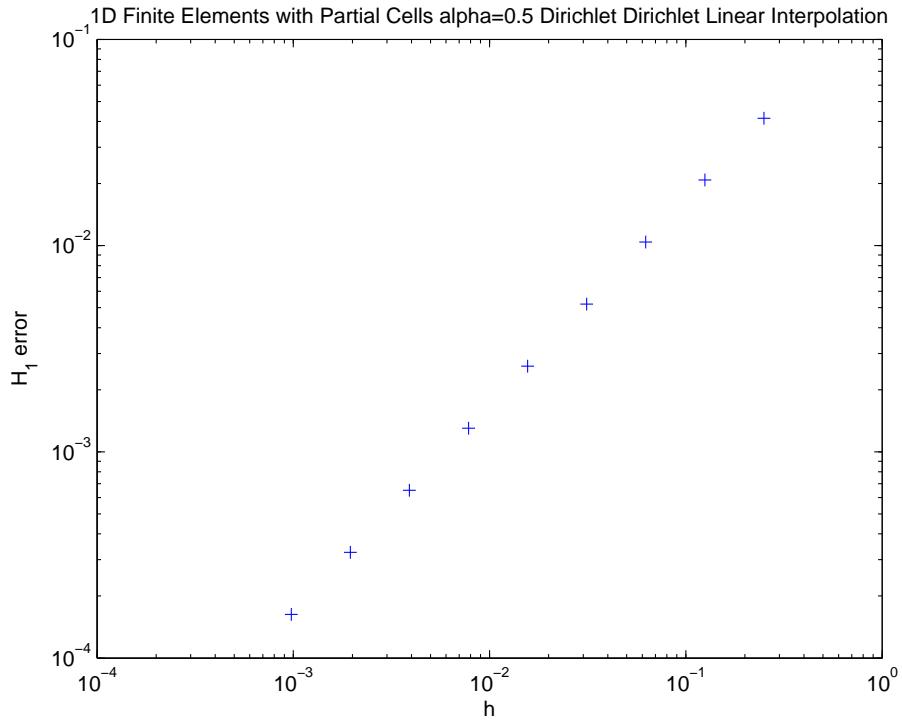


Figure 95: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-convergence

The slope of the convergence curve in the H0-norm is of the order of 1.99. The order of convergence in the H1-norm is unchanged compared to the case of a Dirichlet-Dirichlet-problem. The slope of the convergence curve is of the order of 1.0.

11.2.7 Dirichlet-Neumann-Problem with Finite Elements and Partial Cells 1d

The problem to be solved is:

$$\frac{\partial^2 u}{\partial x^2} = -x \quad \text{on } 0 \leq x \leq 1, \quad (277)$$

$$u|_{x=0} = \bar{u}_1 = 0.0, \quad (278)$$

$$\frac{\partial u}{\partial x}|_{x=1} = \bar{u}_2 = 0.5, \quad (279)$$

$$\alpha = 0.5. \quad (280)$$

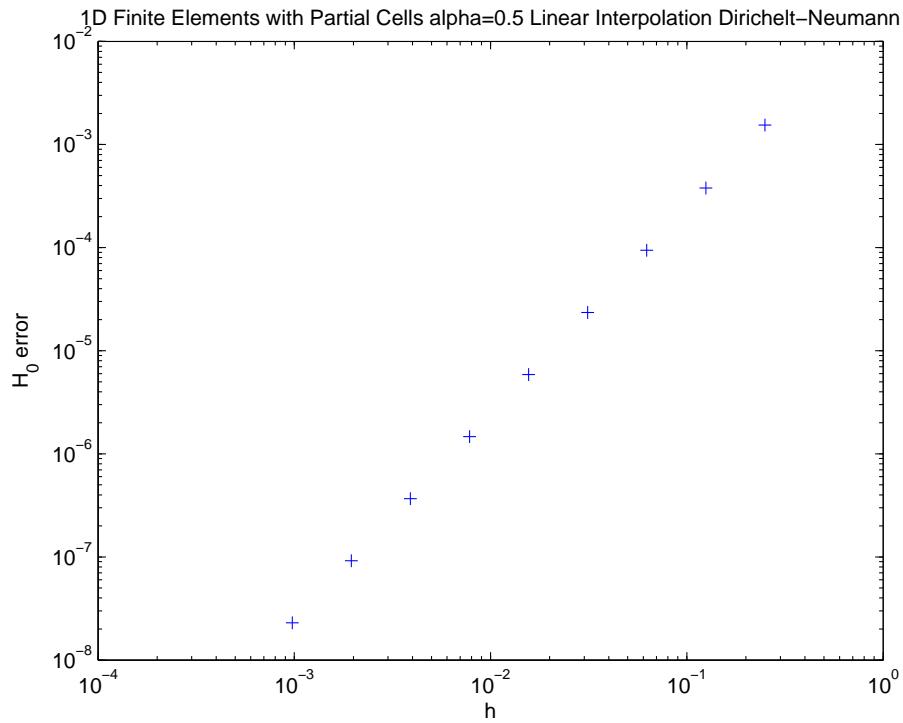


Figure 96: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H₀-convergence

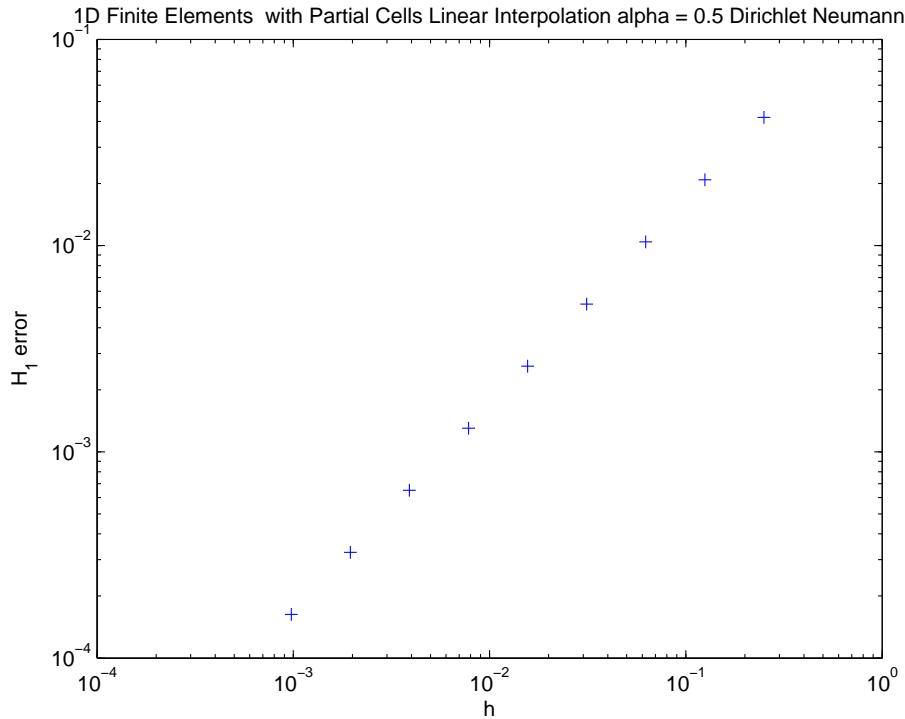


Figure 97: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-convergence

The slope of the convergence curve in the H_0 -norm is of the order of 1.99. One does not notice a loss in the convergence order. The order of convergence in the H_1 -norm is unchanged compared to the case of a Dirichlet-Dirichlet-problem. The slope of the convergence curve is of the order of 1.0.

11.2.8 Dirichlet-Neumann-Problem with Finite Elements without partial cells 2d

I have implemented a heat problem on a rectangular domain with flux and radiation boundary conditions for which I have found the solution in [3]. The canonical problem distracts from the study of the convergence behavior of Neumann boundary problems. The analytical solution found to this problem can not be implemented numerically, because it is given as an infinite sum over a quotient of cosh-functions that blows up when evaluated numerically.

Another problem where upper and lower sides are imposed to constant (homogeneous) but different Dirichlet boundary conditions and the right and left sides are imposed to zero Neumann boundary conditions, the solution is linear and is hit exactly by the finite element solution even for coarser meshes. Therefore, a convergence study is not possible.

In order to create an appropriate canonical problem on a rectangular domain, an analytical solution can be given as green's function which should satisfy the Neumann boundary conditions. I was not able to find such Green's functions in the literature or derive them on my

own.

But, at the end, the convergence study of this example, will lead definitely to a second order convergence in the H0-normal and a first order convergence in the H1-norm.

We defer this study to later, if necessary.

11.2.9 Length of Partial Physical Domain and Convergence Order (1d, Dirichlet Boundary Conditions)

Since the finite volume method and the finite element method in 1d did exhibit the same order of convergence independently of the type of boundary conditions imposed, we want to study the effect of the size of the physical domain with respect to the size of the grid cell on the convergence behavior. In the previous one-dimensional canonical examples solved with the finite element method, the length of the physical domain is exactly half of the cell size. The boundary condition at the end of the physical domain is therefore imposed particularly at the middle of the finite element. The midpoint of a finite element might have particular properties that will simplify or improve the solution. Therefore, we want to vary the studied canonical problem by varying the location of the end of the physical domain with respect to the finite element.

$$\frac{\partial^2 u}{\partial x^2} = 0 \quad \text{on} \quad 0 \leq x \leq 1, \quad (281)$$

$$u|_{x=0} = \bar{u}_1 = 0.0, \quad (282)$$

$$u|_{x=1} = \bar{u}_2 = 0.5 \quad (283)$$

$$(284)$$

For different partial sizes of 0.3, 0.5 and 0.8, the convergence behavior in the H0 and H1 norms for dirichlet-dirichlet boundary problem are given in the figures 98, 99, 100, 101, 102 and 103.

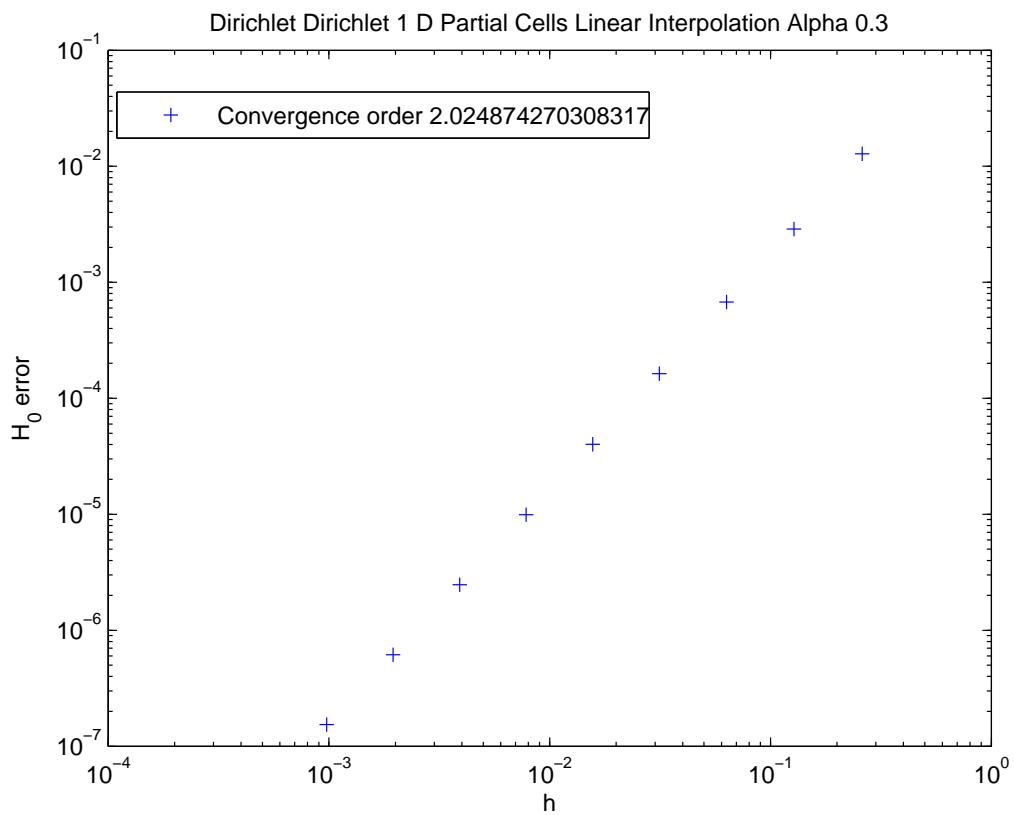


Figure 98: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H₀-convergence, and $\alpha = 0.3$

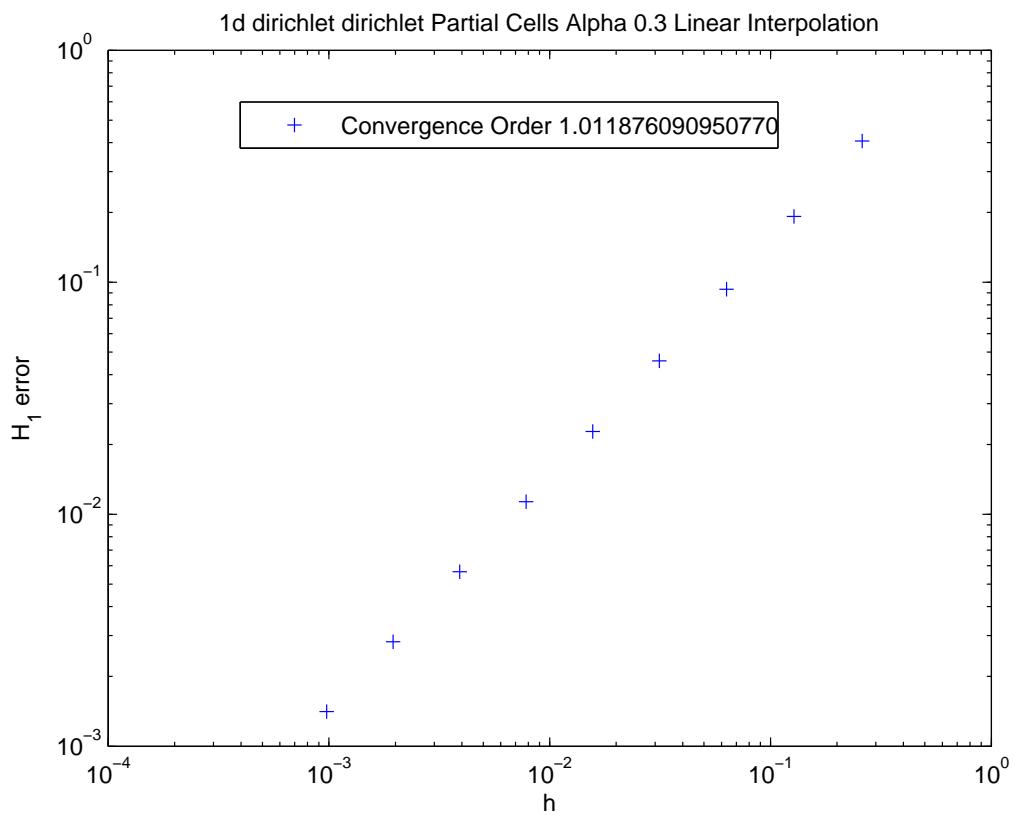


Figure 99: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-convergence, and $\alpha = 0.3$

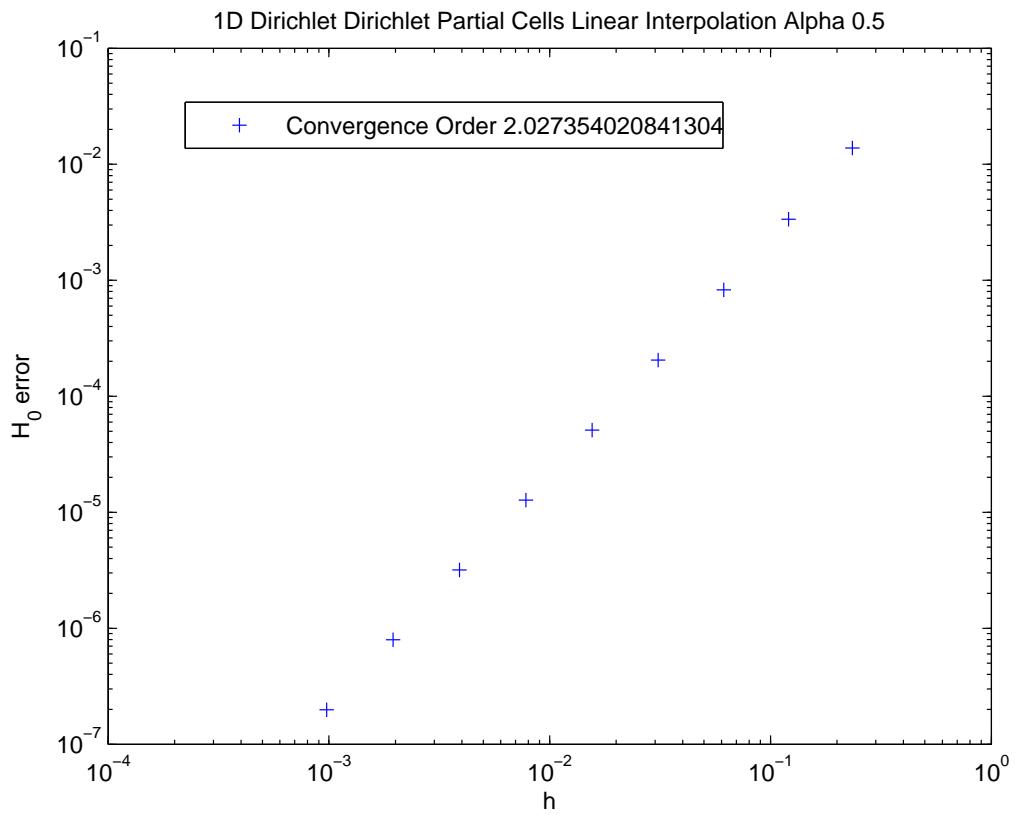


Figure 100: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H₀-convergence, and $\alpha = 0.5$

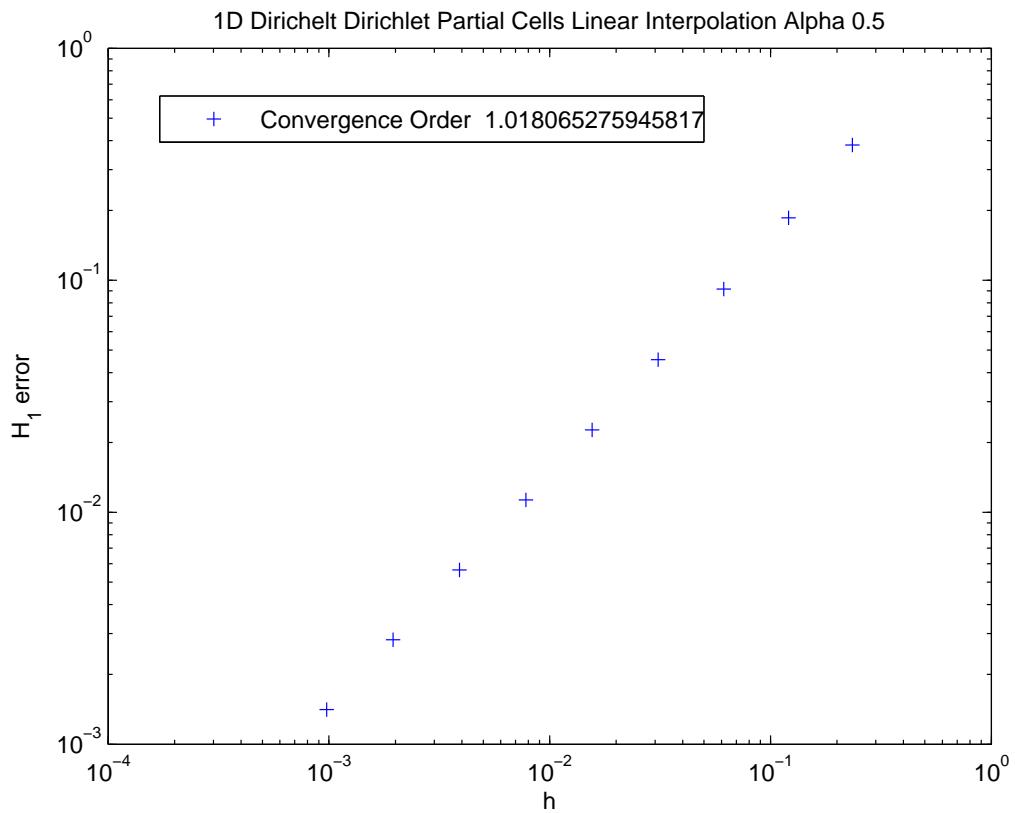


Figure 101: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-convergence, and $\alpha = 0.5$

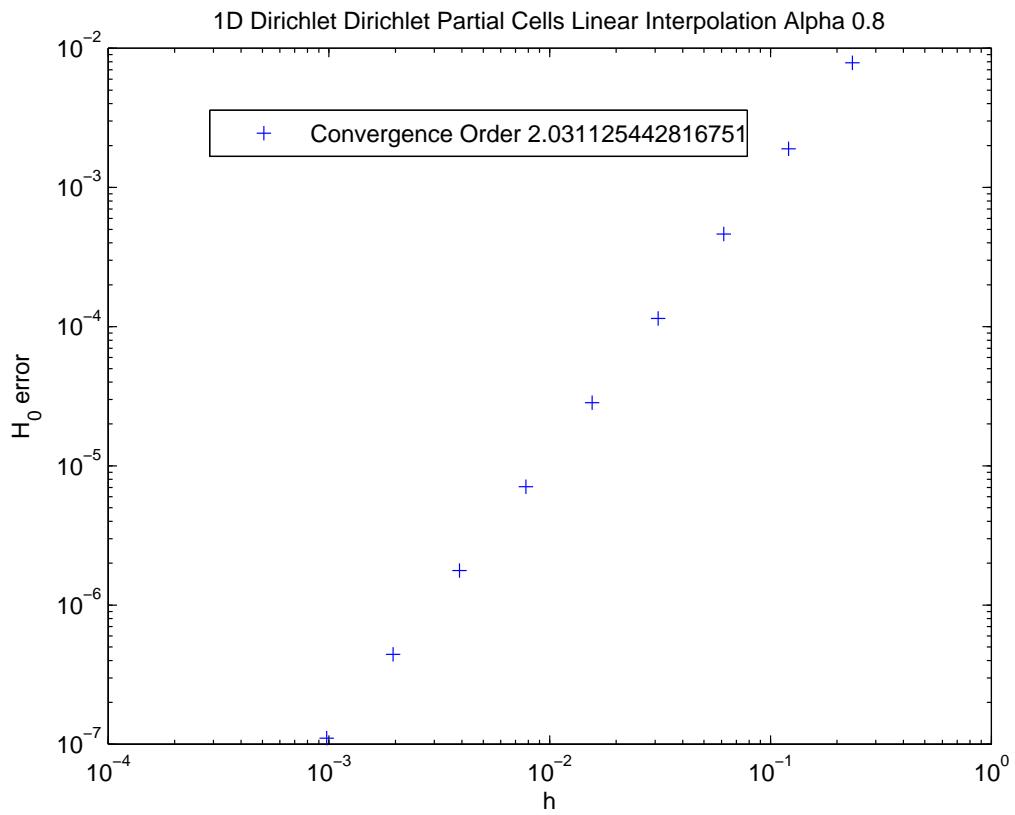


Figure 102: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H₀-convergence, and $\alpha = 0.8$

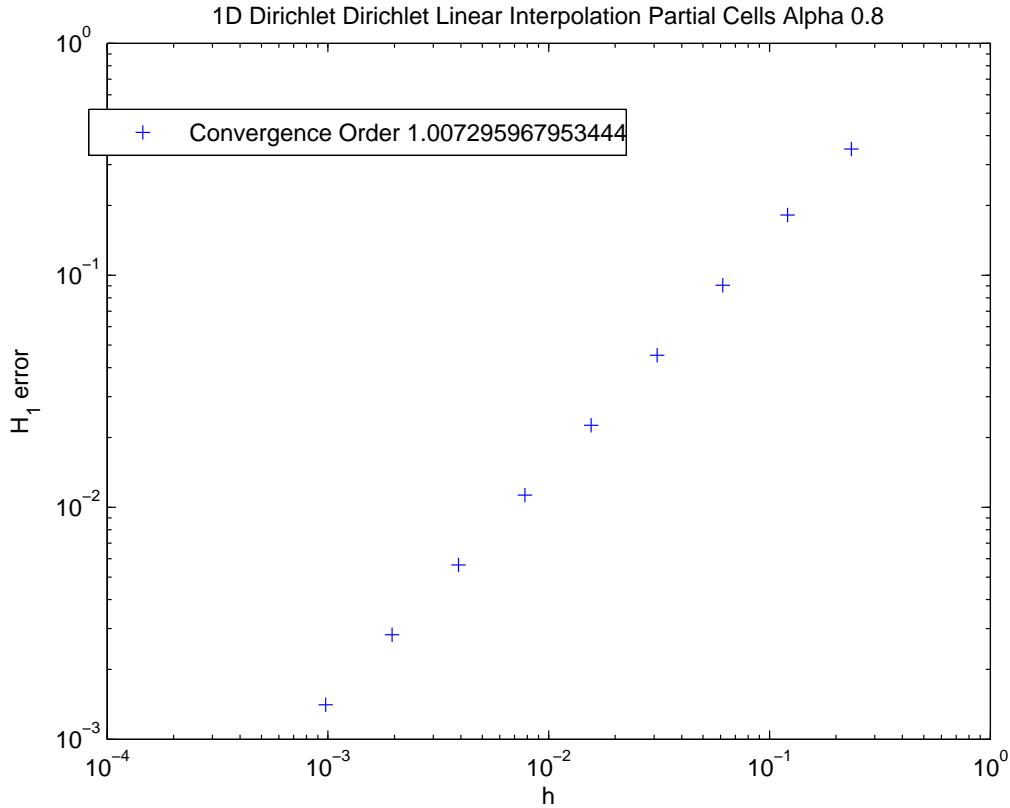


Figure 103: One-dimensional Dirichlet-Dirichlet-Problem solved with Finite Elements and Partial Cells: H1-convergence, and $\alpha = 0.8$

The method of partial cells developed leads to the same convergence order in both norms independently of the relative position of the end of the physical domoain relative to the element.

11.2.10 Length of Partial Cell and Convergence Order (1d, Neumann Boundary Conditions)

$$\frac{\partial^2 u}{\partial x^2} = 0 \quad \text{on} \quad 0 \leq x \leq 1, \quad (285)$$

$$u|_{x=0} = \bar{u}_1 = 0.0, \quad (286)$$

$$\frac{\partial u}{\partial x}|_{x=1} = \bar{u}_2 = 0.5, \quad (287)$$

$$(288)$$

For different partial sizes of 0.3, 0.5 and 0.8, the convergence behavior in the H0 and H1 norms for dirichlet-neumann boundary problem are given in the figures 104, 105, 106, 107, 108 and 109.

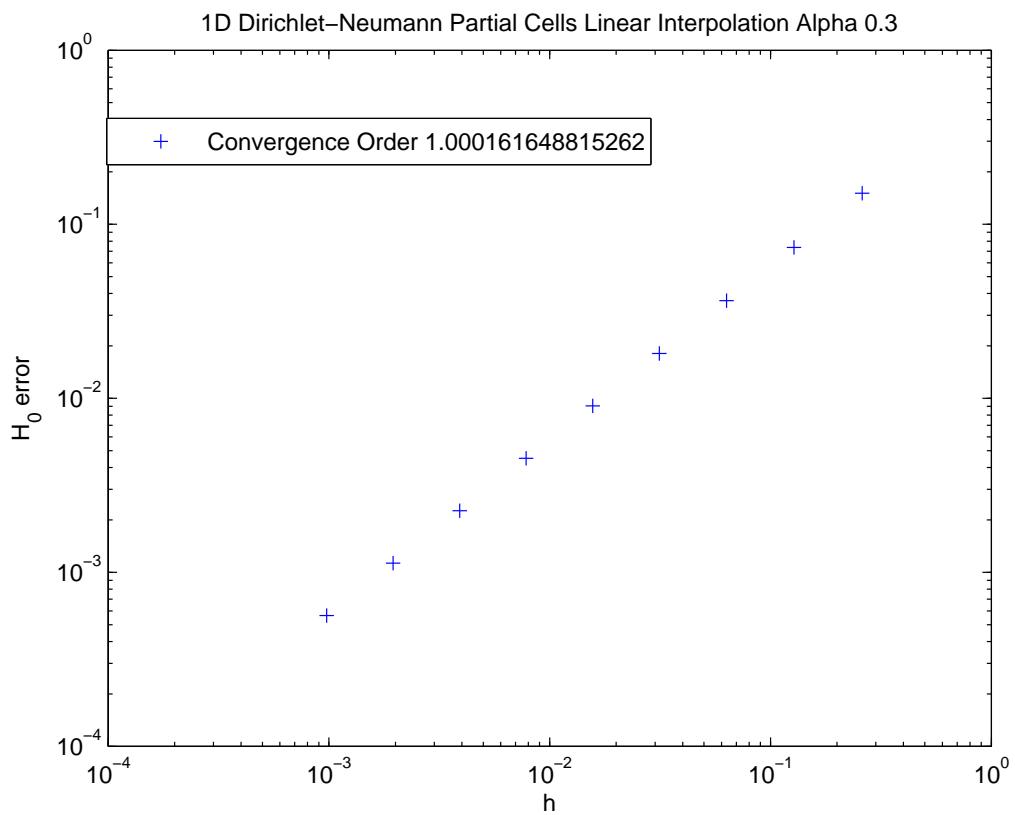


Figure 104: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H_0 -convergence, and $\alpha = 0.3$

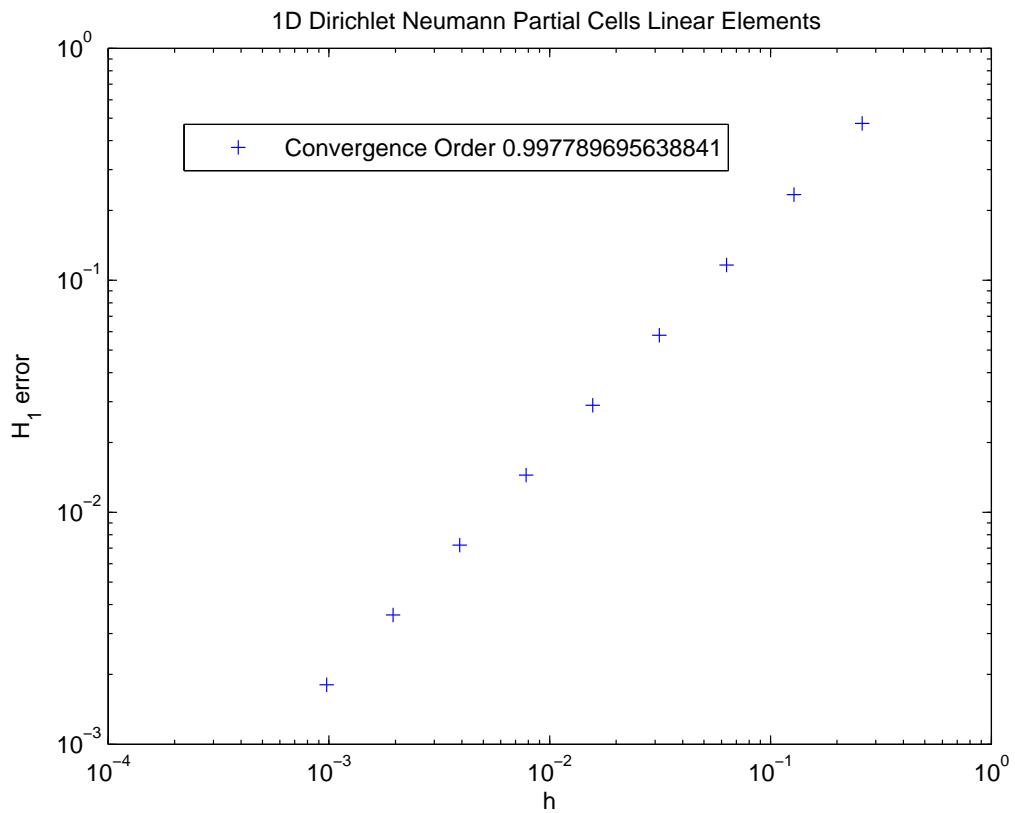


Figure 105: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-convergence, and $\alpha = 0.3$

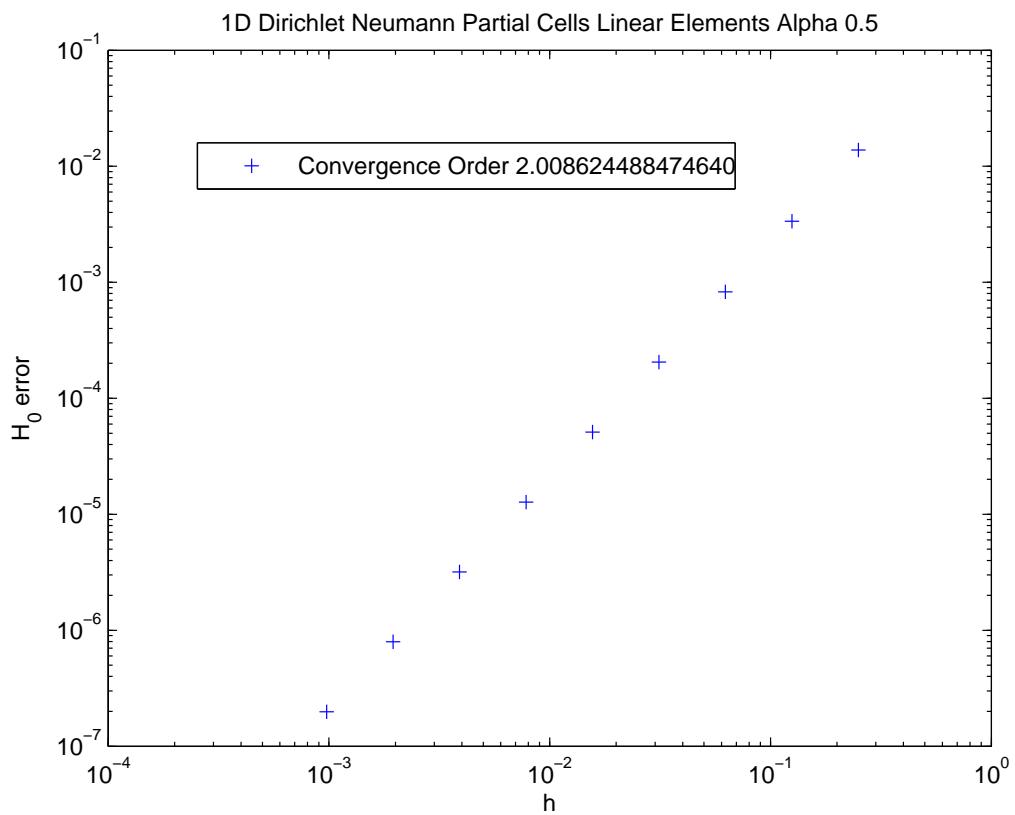


Figure 106: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H₀-convergence, and $\alpha = 0.5$

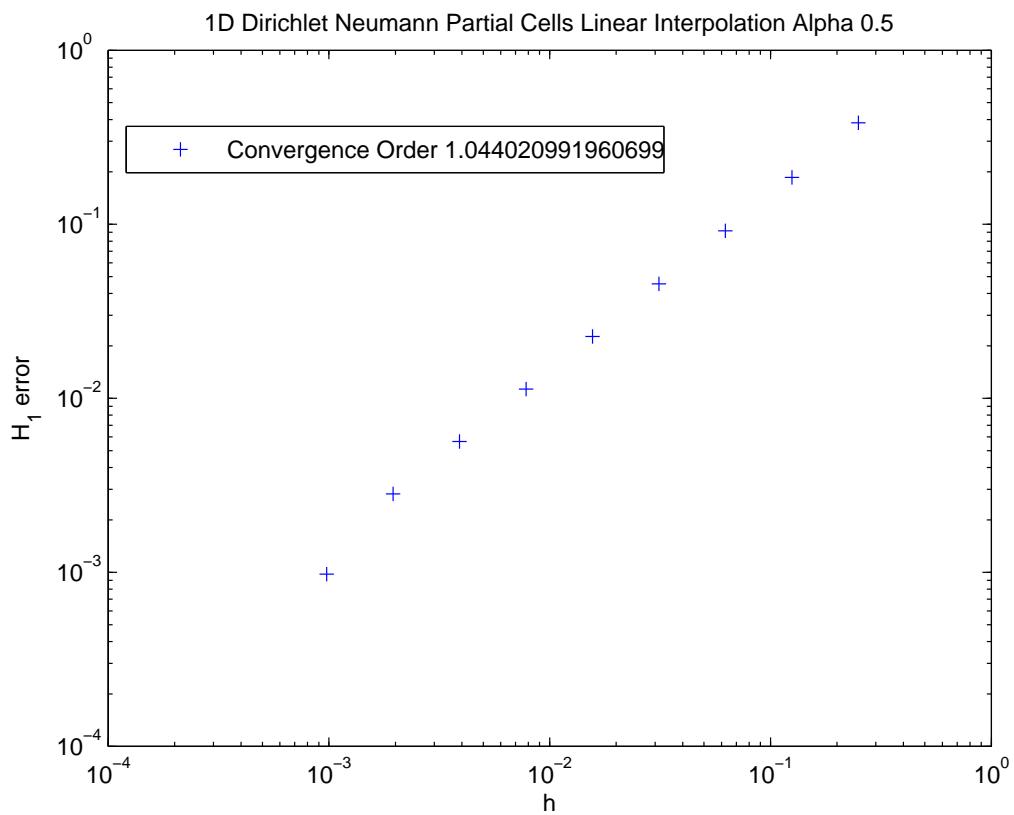


Figure 107: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-convergence, and $\alpha = 0.5$

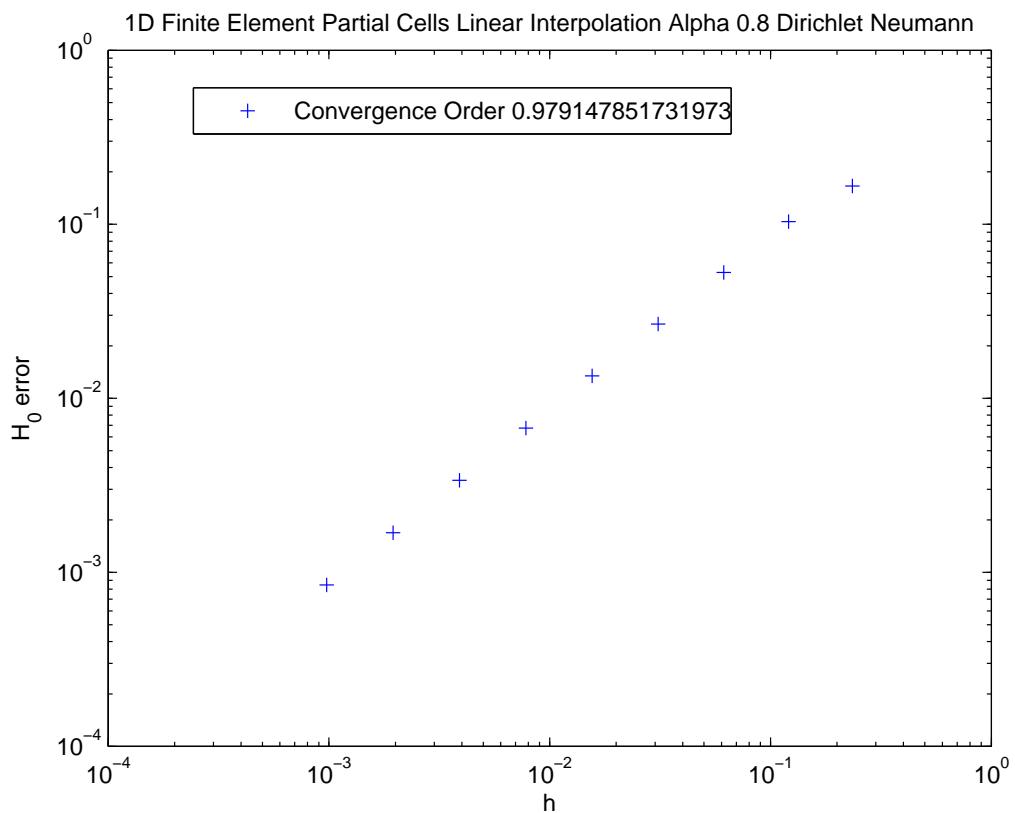


Figure 108: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H₀-convergence, and $\alpha = 0.8$

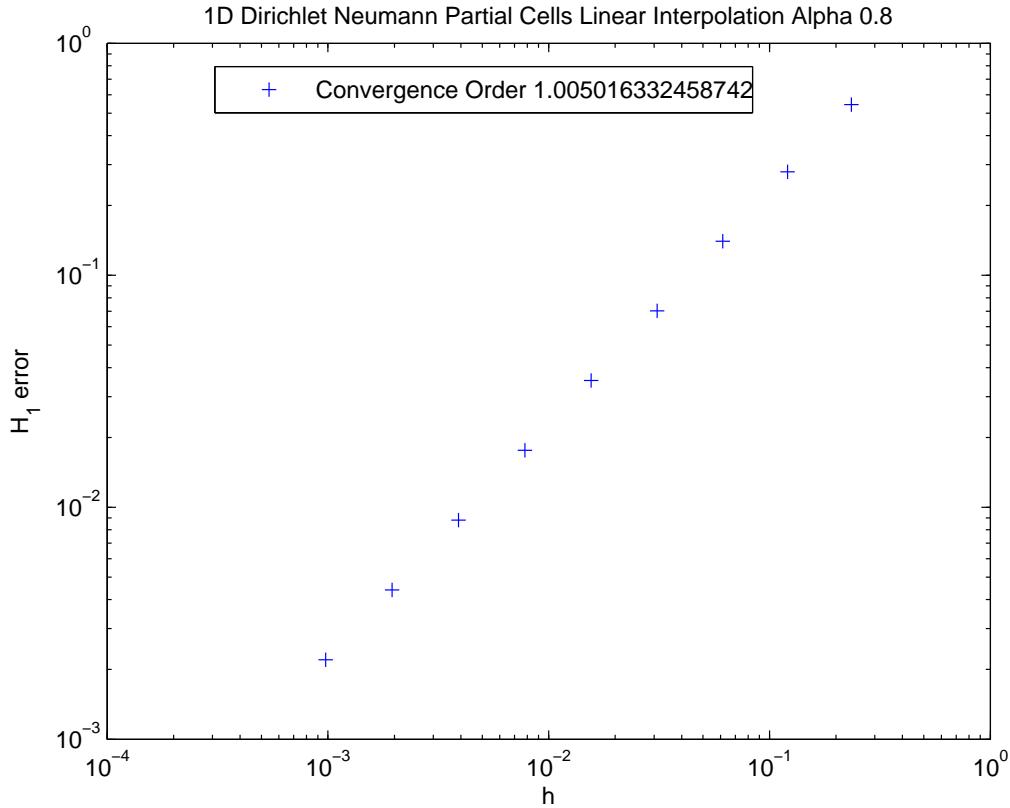


Figure 109: One-dimensional Dirichlet-Neumann-Problem solved with Finite Elements and Partial Cells: H1-convergence, and $\alpha = 0.8$

We notice that the convergence behavior in the H_0 -norm depends on the location of the end of the physical domain when imposing Neumann boundary conditions. The H_1 -error is consistent for all partial element sizes.

11.2.11 Least Squares: Different Method for Imposing Neumann Boundary Conditions

The collocation method used for imposing Neumann boundary conditions has showed a loss in the convergence order due to its dependence on the size of the physical domain in the partial cells. In order to maintain the second order convergence in the H_0 -norm for Neumann boundary problems, we want to use a Least-Squares approximation for imposing the Neumann boundary conditions.

The weak form to the studied heat problem $A \nabla (\cdot \nabla u) = f$ ⁸ is given by:

$$\int_{\partial\Omega} fv \, d\Omega = A \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega - A \int_{\partial\Omega} (\nabla u \cdot \mathbf{n})v \, d\partial\Omega. \quad (289)$$

⁸Sethian: “The heat equation is your mother.”

We want to minimize:

$$\int_{\Omega} (\nabla u \cdot \mathbf{n} - \bar{u}_2)^2 d\Omega, \quad (290)$$

i.e.

$$\int_{\Omega} \nabla \delta u \cdot \mathbf{n} (\nabla u \cdot \mathbf{n} - \bar{u}_2) d\Omega = 0. \quad (291)$$

We add the last term to the weak form, and solve:

$$\int_{\partial\Omega} f v d\Omega - A \int_{\Omega} \nabla u \cdot \nabla v d\Omega + A \int_{\partial\Omega} (\nabla u \cdot \mathbf{n}) v d\partial\Omega + \int_{\Omega} \nabla \delta u \cdot \mathbf{n} (\nabla u \cdot \mathbf{n} - \bar{u}_2) d\Omega = 0. \quad (292)$$

12 Complex Shaped Physical Domain

References

- [1] O.C. Zienkiewicz, R.L. Taylor & J.Z. Zhu, The Finite Element Method, Its basis and fundamentals, Elsevier Butterworth-Heinemann, 2005, pp. 459-465.
- [2] James W. Demmel, Applied Numerical Linear Algebra, Society for Industrial and Applied Mathematics, 1997, pp. 60-62.
- [3] H.S. Carslaw, Conduction of Heat in Solids, Oxford at the Clarendon Press, 1959, pp. 167-168.