Embedded System to Solve the Laplace's Equation

Miguel Ángel San-Pablo-Juárez[1], Eduardo Morales-Sánchez[1], and Fernando Ireta-Moreno[2]

[1] Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada unidad Querétaro, Instituto Politécnico Nacional
miguelangel.sanpablo@gmail.com
[2] División de Ingenierías campus Irapuato-Salamanca, Universidad de Guanajuato
fireta@ugto.mx

Abstract. In this work is solved the Laplace's equation in two dimensions with Dirichlet boundary conditions using the Finite Element Method. Solution algorithms were designed and were loaded into a 32-bit microcontroller STM32. The programmed embedded system may be used in different applications such as solution potential in electrostatics, temperature distribution in heat equation, bubble theory or in image processing. The aim of this work is to solve the Laplace'e equation using an embedded system, instead of using a typical PC.

Keywords: STM32, Laplace's equation, Finite Element Method, Potential

1 Introduction

We present a numerical solution for solving Laplace's equation on two-dimensional irregular regions, using Dirichlet boundary conditions, the solution is implemented on an embedded system that is very fast in time for numerical computations. This approach uses a finite element discretization and avoid the use of a Personal Computer. Solutions for Laplace's equation have been studied in works as[1], based on iteratively solving integral equations; in [2], for mixed boundary value conditions; [3] using a Finite Volume Method and [4] where is used a regularized integral equation method. Studies about Poisson's equation was made too in [5], showing a finite-volume discretization or in [6], where is used a finite difference method. But note that no one have implemented a solution into a microcontroller embedded system yet.

1.1 Laplace's Equation

The partial differential equation that involves $u_{xx}(x, y) + u_{yy}(x, y)$ is called an elliptic equation. A particular elliptic equation we will consider is known as the

Laplace's equation [7]

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = 0 \tag{1}$$

Laplace's equation is often mentioned as a potential equation [8], because the primary task of electrostatics is to find the electric field of a given stationary charge distribution [9] and the best strategy for finding the electric field is first to calculate the potential, V. In this case, equation (1) reduces to Laplace's equation:

$$\boldsymbol{\nabla}^2 V = 0 \tag{2}$$

Where potential function $V(x,y) = u(x,y)$, ∇^2 represents the Laplacian, V the potential, with a charge density $\rho = 0$. This formula is so fundamental to the subject that one might almost say electrostatics *is* the study of Laplace's equation [9]. Some of the most used methods to solve Laplace's Equation are: Separation of Variables, Finite Difference Method and the Finite Element Method.

1.2 The Finite Element Method

The Finite Element Method (FEM) is useful in solving differential equations [10]. The finite element analysis of any problem involves four steps: (a) discretizing the solution region into a finite number of subregions or elements, (b) deriving governing equations for a typical element, (c) assembling of all elements in the solution region, and (d) solving the system of equations obtained [10]. FEM is the most commonly method used to solve the problem, as it is very easy to apply to problems with any type geometries [7].

Finite Element Discretization The solution region is divided into a number of finite elements as in Fig. 1, where the region is subdivided into six elements (all triangular) and seven nodes. We seek an approximation for the potential V_e within an element e and then interrelate the potential distributions in various elements such that the potential is continuous across interelement boundaries [10]. The approximate solution for the whole region is [7]:

$$V(x,y) \simeq \sum_{e=1}^{N} V_e(x,y) \tag{3}$$

where N is the number of triangular elements into which the solution region is divided. The most used approximation for V_e within an element is a polynomial approximation of the form [7]

$$V_e(x,y) = a + bx + cy \tag{4}$$

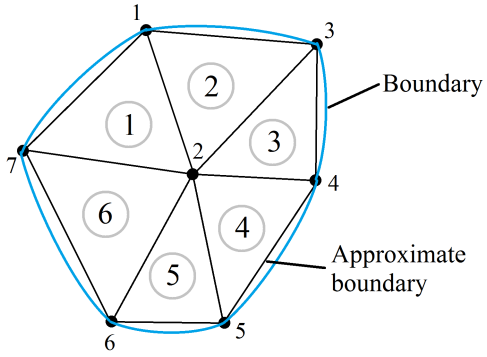

Fig. 1. A typical finite element subdivision used for an irregular domain

in this case for a triangular element. Notice that our assumption of linear variation of potential within the triangular element as in equation (4) is the same assuming that the electric field is uniform within the element; that is, [10]

$$\mathbf{E}_e = -\boldsymbol{\nabla} V_e = -(b\mathbf{a}_x + c\mathbf{a}_y) \tag{5}$$

Element Governing Equations Consider a triangular element as shown in Fig. 2, The potential V_{e1}, V_{e2} and V_{e3} at nodes 1, 2 and 3 respectively, are obtained using equation (4)

$$\begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{6}$$

And the coefficients a, b and c can be determined from equation (6)

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix} \tag{7}$$

Equation (4) can be rewritten as

$$V_e = \begin{bmatrix} 1 & x & y \end{bmatrix} \frac{1}{2A} \begin{bmatrix} (x_2y_3 - x_3y_2) & (x_3y_1 - x_1y_3) & (x_1y_2 - x_2y_1) \\ (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_2) & (x_1 - x_3) & (x_2 - x_1) \end{bmatrix} \begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix} \tag{8}$$

where A is the area of the element e. Then (8) can be expressed as

$$V_e = \sum_{i=1}^{3} N_i(x, y) V_{ei} \tag{9}$$

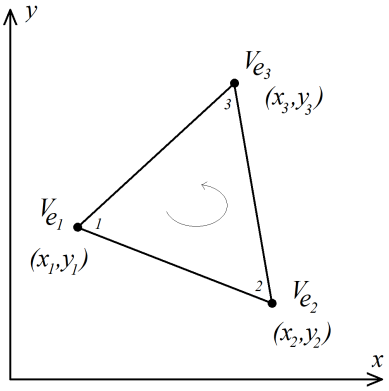

Fig. 2. A typical triangular element. The local numbering could be as 1-2-3.

where N_i represent linear interpolating functions. They are called *element shape functions*.

Now, the energy per unit length associated with the element e is given by [10,9]:

$$W_e = \frac{\varepsilon}{2} \int \mid \mathbf{E} \mid^2 dS = \frac{\varepsilon}{2} \int \mid \boldsymbol{\nabla} V_e \mid^2 dS \tag{10}$$

assuming a two dimensional solution region free of charge ($\rho_S = 0$). Now if it is applied the gradient to (9) this is

$$\boldsymbol{\nabla} V_e = \sum_{i=1}^{3} V_{ei} \boldsymbol{\nabla} N_i \tag{11}$$

and substituting (11) into (10)

$$W_e = \frac{\varepsilon}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} V_{ei} \left[\int \nabla N_i \cdot \nabla N_j dS \right] \tag{12}$$

If is defined the term in brackets as

$$C_{ij}^{(e)} = \int \boldsymbol{\nabla} N_i \cdot \boldsymbol{\nabla} N_j dS \tag{13}$$

equation (12) becomes

$$W_e = \frac{\varepsilon}{2} \left[V_e \right]^T \left[C^{(e)} \right] \left[V_e \right] \tag{14}$$

Asembling all elements Finally is needed to assemble each element in the solution region, for this, the energy associated with the assemblage of all elements is required

$$W = \sum_{e=1}^{N} W_e = \frac{1}{2}\varepsilon[V]^T[C][V] \tag{15}$$

with

$$[V] = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \tag{16}$$

N is the number of elements, n is the number of nodes and $[C]$ is the *global coefficient matrix*, that is the assemblage of each individual element coefficient matrices.

$$[C] = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1\,41} \\ C_{21} & C_{22} & \cdots & C_{2\,41} \\ \vdots & \vdots & \ddots & \vdots \\ C_{41\,1} & C_{41\,2} & \cdots & C_{41\,41} \end{bmatrix} \tag{17}$$

Here C_{ij} is obtained with the fact that the potential distribution must be continuous across interelement boundaries. The size of C depends on the amount of involved nodes and C_{ij} is the coupling between nodes i and j. Some properties of the C_{ij} matrix are:

- Is symmetric ($C_{ij} = C_{ji}$) as the element coefficient matrix
- Since $C_{ij} = 0$ if does not exist coupling between nodes i and j, for a large number of elements $[C]$ becomes sparse and banded.
- It is singular.

Solving the Resulting Equations From variational calculus, it is known that Laplace's equation is satisfied when the total energy in the solution region is minimum. We require that the partial derivatives of W with respect to each nodal value of the potential be zero. Using the Band Matrix Method together with Gauss-Jordan method it is possible to minimize the total energy in the solutions region (derivatives of W respect each nodal value of the potential be zero). Finally the vector solution of equation (16) can be found and a potential distribution with equation (9) may be obtained. Because of a separation of fixed and free nodes in the Band Matrix method, free element coefficient matrix is not singular and it is possible to inverse for finding free node potentials. Once analytically analyzed the problem, an algorithm for solving the problem can be designed to implement into the STM32 microcontroller. Fig. 3 shows the solution algorithm to solve Laplace's equation in a two dimensions region with Dirichlet boundary conditions.

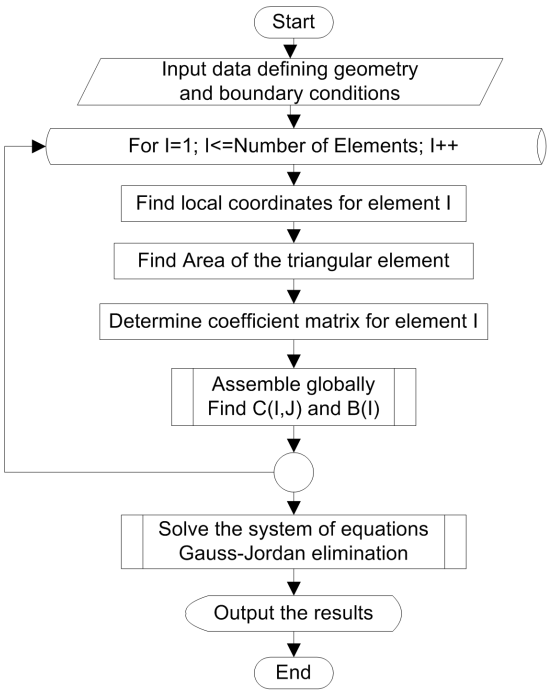

Fig. 3. Algorithm to solve Laplace's equation.

1.3 Implementation on the STM32F Discovery

Laplace's equation solution by the FEM is typically made using a PC, with solution algorithms implemented in compilers of different high-level languages or using specialized commercial software. But some specific applications require calculations programmed into embedded systems. Examples are: Electrostatics, Electrical Impedance Tomography, temperature measurements, image analysis an so on. A big advantage to solve a differential equation in an embedded system is the reduction of space in the final implementation. Another advantage is that the system performs only a particular task [11], in this case the solution of an elliptic differential equation. Power consumption and portability are other advantages.

In this work, a microcontroller (MCU) of the family STM32 32-bit ARM Cortex was used. The STM32 family of 32-bit Flash microcontrollers are based on the ARM©Cortex©-M processor. It offers a 32-bit product range that combines very high performance, real-time capabilities, digital signal processing, and low-power, low-voltage operation, while maintaining full integration and ease of development [12]. Such computational characteristics are sufficient to implement the algorithm shown in Fig. 3.

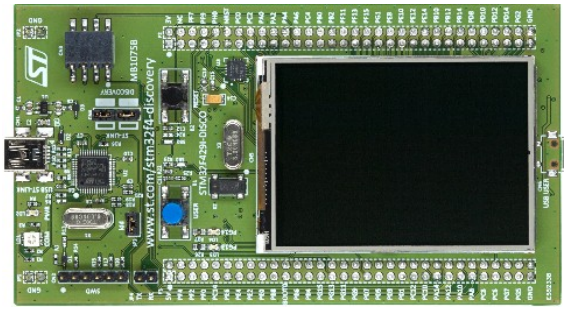

Fig. 4. Embedded system STM32F429IDiscovery where solution was loaded [13]

Fig. 4 shows a target with MCU STM32F429ZIT6 and a LCD to display numeric results. This kind of board is the embedded system used to implement the solution.

2 Results

The development of the source code has four stages:

– First, the input of the number of elements, the number of nodes, the number of fixed nodes, the prescribed values of the potentials at the free nodes, the

x and y coordinates, element-node relationship and prescribed potentials at fixed nodes.
- Second, to find the element coefficient matrix $[C^{(e)}]$ for each element and the global coefficient matrix $[C]$.
- Third, to solve the inverse of the gobal matrix by Gauss-Jordan elimination technique.
- Finally, outputting the result of the computation on a screen or display.

Numerical results are shown by an electrostatic application example.

2.1 Validation with an electrostatic example

The algorithm that solve Laplace's equation was compiled in a free version of Keil uVision 5 and loaded into a STM32F429IDiscovery board. For test the embedded system, it was used a triangle region example from [10] that consists in a two-dimensional problem as shown in Fig. 5. The triangular region is divided into 25 triangular elements with a total of 21 nodes and some Dirichlet boundary conditions are taken as: $100V$ in nodes 11, 15, 18 and 20; $50V$ in nodes 6 and 21; $0V$ in nodes 1, 2, 3, 4, 5, 6 and same $0V$ in nodes 7, 12, 16, 19.

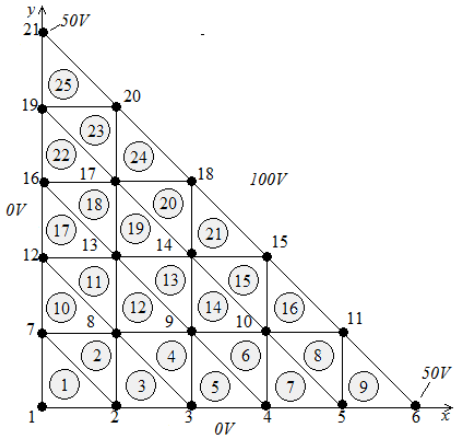

Fig. 5. Solution region divided into 25 elements

The numerical solution is computed in each node and displayed into STM32F Discovery board screen as is shown in Fig. 6. It is possible to observe a numerical comparison with results of the same example from [10] in Table 1.

An important observation is that Discovery board can process operations with matrices of size up 5 x 5 without problems; with operations between matrix

Fig. 6. STM32F running an example

Node No.	Reference Potential	Computed Potential	Error
1	0.000	0.0000	0.0000
2	0.000	0.0000	0.0000
3	0.000	0.0000	0.0000
4	0.000	0.0000	0.0000
5	0.000	0.0000	0.0000
6	50.000	50.0000	0.0000
7	0.000	0.0000	0.0000
8	18.182	18.1818	0.0002
9	36.364	36.3636	0.0004
10	59.091	59.0909	0.0001
11	100.000	100.0000	0.0000
12	0.000	0.0000	0.0000
13	36.364	36.3636	0.0004
14	68.182	68.1818	0.0002
15	100.000	100.0000	0.0000
16	0.000	0.0000	0.0000
17	59.091	59.0909	0.0001
18	100.000	100.0000	0.0000
19	0.000	0.0000	0.0000
20	100.000	100.0000	0.0000
21	50.000	50.0000	0.0000

Table 1. Comparison between reported potential and computed potential.

sizes of 90 x 90 can be observed that there are some processing problems, and for sizes for more than 100 x 100 it is needed to add auxiliary variables when we compile in Keil free version compiler. The auxiliary variable can be a simple double type or a double vector type. The syntax is:

```
for(...){
...
auxiliary[i]=C[i]
C[i] = auxiliary[i] + value[i]*matrix[i][j];
...
}
```

instead of

```
for(...){
...
C[i] = C[i] + value[i]*matrix[i][j];
...
}
```

3 Conclusions

It is possible to solve Laplace's equation using a microcontroller of the family STM32 and avoiding the typical use of a PC. Both the global matrix compute and the Gauss-Jordan elimination are supported and solved very fast. To solve large vector and matrix operations such as addition in the Keil compiler, it is needed an auxiliary variable to compute correctly operations of recursive type $variable[i] = variable[i] + function(i)$; using this form: $auxiliary[i] = variable[i]$; and $variable[i] = auxiliary[i] + function(i)$;.

Acknowledgments The authors would like to express their gratitude to Instituto Politécnico Nacional for the support given in SIP 20144683 and SIP 20151350 projects. Thanks to CONACYT for scholarship (CVU/Scholarship holder) 350051/237150.

References

1. Rokhlin, V.: Rapid solution of integral equations of classical potential theory. Journal of Computational Physics. 60, 187–207 (1985).
2. Wendland, W.L., Stephan, E., Hsiao, G.C., Meister, E.: Mathematical Methods in the Applied Sciences. 1, 265–321 (1979).
3. Komla, D., Pascal, O.: A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. ESAIM: Mathematical Modelling and Numerical Analysis. 39, 1203-1249 (2005).

4. Liu, C.S.: A Meshless Regularized Integral Equation Method for Laplace Equation in Arbitrary Interior or Exterior Plane Domains. CMES. 19, 99–109 (2007).
5. Johansen, H. , Colella, P.: A Cartesian Grid Embedded Boundary Method for Poissons Equation on Irregular Domains. Journal of Computational Physics. 147, 60–85 (1998).
6. Jomaa, Z., Macaskill, C.: The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions. Journal of Computational Physics. 202, 488–506 (2005).
7. Burden, R.L., Faires, J.D.: Numerical Analysis. Cengage Learning, (2011).
8. Boyce, W.E., DiPrima, R.C.: Elementary Differential Equations and Boundary Value Problems. Wiley, (2013).
9. Griffiths, D.J.: Introduction to electrodynamics. Prentice Hall, (1999).
10. Sadiku, M.O.: Elements of Electromagnetics. Oxford, (2004).
11. Heath, S.: Embedded Systems Design. Newnes, (2003).
12. STMicroelectronics, `http://www.st.com/web/en/catalog/mmc/FM141/SC1169`, 05-05-2015.
13. STMicroelectronics, `http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1848/PF259090?s_searchtype=partnumber`, 01-08-2015