Rendiconti Sem. Mat. Univ. Pol. Torino
Vol. 76, 2 (2018), 77 – 86

R. Cavoretto* **- A. De Rossi***

AN ADAPTIVE ALGORITHM BASED ON RBF-PU COLLOCATION FOR SOLVING 2D POISSON PROBLEMS

Abstract.

In this paper we present a new adaptive algorithm to solve 2D Poisson problems via a radial basis function partition of unity collocation method. The refinement process depends on an error indicator, which consists in comparing two approximate solutions computed on a coarser set and a finer one of collocation points. This technique is able to detect the domain areas that need to be refined, so that the accuracy of the method can be improved. Numerical results illustrate the performance of the adaptive algorithm.

1. Introduction

In this work we investigate the problem of constructing adaptive methods via radial basis functions (RBFs) for solving 2D Poisson problems. In particular, the use of adaptive meshless algorithms has recently been considered by applying different RBF-based approaches, which include finite difference (RBF-FD) [4, 10], partition of unity (RBF-PU) [2] or, more in general, RBF collocation [5, 9] methods. More precisely, in [2] we proposed within a RBF-PU framework a new error estimator based on the comparison of two collocation solutions evaluated on a coarser set and a finer one. Since this indicator was not extensively analyzed in our previous paper, here we present in detail the new adaptive algorithm along with its error indicator. This study is further supported by numerical experiments carried out on some test problems.

The paper is organized as follows. In Section 2 we review some basic information of the RBF-PU collocation method for solving elliptic PDE problems. In Section 3 we describe the adaptive algorithm along with its refinement startegy. In Section 4 we show some numerical results carried out to illustrate the performance of our adaptive scheme. Section 5 contains conclusions and future work.

2. RBF-PU collocation method

Let Ω be an open and bounded domain on $\mathbb{R}^s$ and $u : \Omega \to \mathbb{R}$. Given a set $X_N = \{x_1, \ldots, x_N\} \subseteq \Omega$ of collocation points, we partition the domain Ω into a number of d subdomains or patches Ω_j such that $\bigcup_{j=1}^{d} \Omega_j \supseteq \Omega$ with some mild overlaps among them [8]. As subdomains we take hyperspherical patches on $\mathbb{R}^s$ of fixed radius such that $\{\Omega_j\}_{j=1}^{d}$ is a covering of Ω. Along with the subdomains we define a partition of unity by considering a family of compactly supported, nonnegative and continuous

*Member of the INdAM Research group GNCS.

RBF	$\phi_\varepsilon(r)$
Inverse MultiQuadric C^∞ (IMQ)	$(1+\varepsilon^2 r^2)^{-1/2}$
MultiQuadric C^∞ (MQ)	$(1+\varepsilon^2 r^2)^{1/2}$
Mat é rn C^6 (M6)	$e^{-\varepsilon r}(\varepsilon^3 r^3+6\varepsilon^2 r^2+15\varepsilon r+15)$
Mat é rn C^4 (M4)	$e^{-\varepsilon r}(\varepsilon^2 r^2+3\varepsilon r+3)$

Table 1: Some examples of popular RBFs table 1, 2 and 3 (also in the text).

functions w_j, with $\text{supp}(w_j) \subseteq \Omega_j$, and such that

$$\sum_{j=1}^{d} w_j(x) = 1, \qquad \forall x \in \Omega.$$

A possible choice is given by *Shepard's weight*

$$w_j(x) = \frac{\tilde{w}_j(x)}{\sum_{k=1}^{d} \tilde{w}_k(x)}, \qquad j = 1,\dots,d,$$

where $\tilde{w}_j$ are compactly supported functions on Ω_j (see [3, 11]).

After choosing the weights $\{w_j\}_{j=1}^d$, the global RBF-PU approximant is formed by a weighted sum of local RBF approximants $\tilde{u}_j$

$$\tilde{u}(x) = \sum_{j=1}^{d} w_j(x)\tilde{u}_j(x), \qquad x \in \Omega. \tag{2.1}$$

with

$$\tilde{u}_j(x) = \sum_{i=1}^{N_j} c_i^j \phi_\varepsilon(||x - x_i^j||_2), \tag{2.2}$$

where N_j identifies the number of nodes in Ω_j, i.e. the points $x_i^j \in X_{N_j} = X_N \cap \Omega_j$, c_i^j is an unknown real coefficient, $||\cdot||_2$ denotes the Euclidean norm, and $\phi : \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ is a RBF depending on a positive *shape parameter* ε such that

$$\phi_\varepsilon(||x-z||_2) = \phi(\varepsilon||x-z||_2), \qquad \forall x, z \in \Omega.$$

In Table 1 we list some examples of RBFs with their smoothness degrees (see e.g. [1, 12] for details). These radial functions are commonly used for solving PDEs.

Now, given a linear elliptic differential operator $\mathcal{L}$, our aim is to find the approximate solution of the elliptic PDE problem equipped with Dirichlet boundary conditions

$$\begin{aligned} -\mathcal{L}u(x) &= f(x), \qquad x \in \Omega, \\ u(x) &= g(x), \qquad x \in \partial\Omega. \end{aligned} \tag{2.3}$$

The problem (2.3) is then discretized on the global set X_N of collocation points, which is split into a set X_{N_I} of interior points and a set X_{N_B} of boundary points such that $X_N = X_{N_I} \cup X_{N_B}$, where N_I and N_B denote the number of interior and boundary discretization points, respectively.

Assuming that the problem (2.3) has an approximate solution of the form (2.1) (see [7]), we get

$$\begin{aligned} -\mathcal{L}\tilde{u}(x_i) &= -\sum_{j=1}^{d} \mathcal{L}\left(w_j(x_i)\tilde{u}_j(x_i)\right) = f(x_i), \qquad x_i \in \Omega \\ \tilde{u}(x_i) &= \sum_{j=1}^{d} w_j(x_i)\tilde{u}_j(x_i) = g(x_i), \qquad x_i \in \partial\Omega. \end{aligned} \tag{2.4}$$

Focusing on a Laplace problem, i.e. taking $\mathcal{L} = -\Delta$, the elliptic operator $\mathcal{L}$ can expanded as

$$-\Delta\left(w_j(x_i)\tilde{u}_j(x_i)\right) = -\Delta w_j(x_i)\tilde{u}_j(x_i) - 2\nabla w_j(x_i)\cdot\nabla\tilde{u}_j(x_i) - w_j(x_i)\Delta\tilde{u}_j(x_i), \quad x_i \in \Omega. \tag{2.5}$$

Defined then the vector $\tilde{u}_j = (\tilde{u}_j(x_1^j), \ldots, \tilde{u}_j(x_{N_j}^j))^T$ of local nodal values, we know that the local coefficient vector $c_j = (c_1^j, \ldots, c_{N_j}^j)^T$ is such that $c_j = A_j^{-1}\tilde{u}_j$. So we obtain

$$\Delta\tilde{u}_j = A_j^{\Delta}A_j^{-1}\tilde{u}_j, \qquad \nabla\tilde{u}_j = A_j^{\nabla}A_j^{-1}\tilde{u}_j, \tag{2.6}$$

where A_j^{Δ} and A_j^{∇}, $j = 1, \ldots, N_j$, are the matrices with entries

$$(A_j^{\Delta})_{ki} = \Delta\phi(||x_k^j - x_i^j||_2),$$

and

$$(A_j^{\nabla})_{ki} = \nabla\phi(||x_k^j - x_i^j||_2).$$

Associated with each subdomain Ω_j, we consider the following diagonal matrices

$$\begin{aligned} W_j^{\Delta} &= \operatorname{diag}\left(\Delta w_j(x_1^j), \ldots, \Delta w_j(x_{N_j}^j)\right), \\ W_j^{\nabla} &= \operatorname{diag}\left(\nabla w_j(x_1^j), \ldots, \nabla w_j(x_{N_j}^j)\right), \\ W_j &= \operatorname{diag}\left(w_j(x_1^j), \ldots, w_j(x_{N_j}^j)\right). \end{aligned}$$

In order to get the discrete operator P_j, we differentiate (2.4) by using a product derivative rule and then apply the relations given in (2.6). Making use of (2.5) and including the boundary conditions, the discrete local Laplacian can thus be expressed as follows

$$(P_j)_{ki} = \begin{cases} (\bar{P}_j)_{ki}, & x_i^j \in \Omega, \\ \delta_{ki}, & x_i^j \in \partial\Omega, \end{cases}$$

where δ_{ki} is the Kronecker delta and

$$\tilde{P}_j = \left(W_j^{\Delta} A_j + 2W_j^{\nabla} \cdot A_j^{\nabla} + W_j A_j^{\Delta} \right) A_j^{-1}.$$

By assembling the local matrices P_j into the global matrix P, i.e.

$$(P_j)_{ki} = \sum_{j=1}^{d} (P_j)_{\eta_{kj},\eta_{ij}}, \qquad k,i = 1,\ldots,N,$$

we can derive the global discrete operator, and finally solve the (global) sparse linear system of the form

$$Py = u, \tag{2.7}$$

where $u = (u_1,\ldots,u_N)^T$ is defined by

$$u_i = \begin{cases} f(x_i), & x_i \in \Omega, \\ g(x_i), & x_i \in \partial\Omega. \end{cases}$$

and the numerical solution $y = (\tilde{u}(x_1),\ldots,\tilde{u}(x_N))^T$ is obtained by inverting the collocation matrix P in (2.7).

3. Adaptive algorithm

In this section we present the iterative algorithm proposed to adaptively solve the PDE problem via the RBF-PU collocation method. It can be summarized as follows.

`Step 1`. We define two sets X_N of grid collocation points of size $N_1^{(0)}$ and $N_2^{(0)}$, such that $N_1^{(0)} < N_2^{(0)}$ (the symbol $^{(0)}$ identifies the initial phase, or more in general the iteration). To make clearer the description, we denote these sets as $X_{N_1^{(0)}}$ and $X_{N_2^{(0)}}$, respectively.

`Step 2`. For $k = 0,1,\ldots$, we iteratively determine the two collocation solutions of the form (2.1), i.e. $\tilde{u}_{N_1^{(k)}}$ and $\tilde{u}_{N_2^{(k)}}$, computed on $N_1^{(k)}$ and $N_2^{(k)}$ collocation points, respectively.

`Step 3`. We compare the two approximate solutions by evaluating the error on the (coarser) set containing $N_1^{(k)}$ points, i.e.

$$\left| \tilde{u}_{N_2^{(k)}}(x_i) - \tilde{u}_{N_1^{(k)}}(x_i) \right|, \qquad x_i \in X_{N_1^{(k)}}.$$

Note that here we suppose that the solution given with $N_2^{(k)}$ collocation points is more accurate than that containing $N_1^{(k)}$ points only.

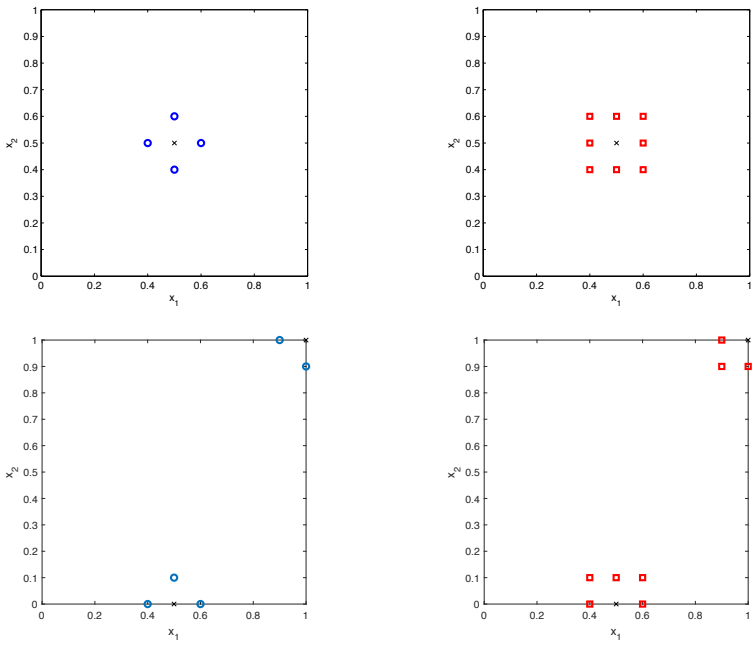

Figure 1: Examples of refinement strategy regarding the sets $X_{N_1^{(k)}}$ (left) and $X_{N_2^{(k)}}$ (right) in the adaptive algorithm.

`Step 4`. Fixed a tolerance tol, we find all points $x_i \in X_{N_1}^{(k)}$ such that

$$|\tilde{u}_{N_2(k)}(x_i) - \tilde{u}_{N_1(k)}(x_i)| > tol. \tag{3.8}$$

`Step 5`. To refine the data point distribution, we compute the *separation distance*

$$q_{X_{N_1}^{(k)}} = \frac{1}{2} \min_{i \neq j} ||x_i - x_j||_2, \qquad x_i \in X_{N_1^{(k)}}. \tag{3.9}$$

`Step 6`. For $k = 0, 1, \ldots$ we update the two sets $X_{N_1^{(k+1)}}$ and $X_{N_2^{(k+1)}}$ of discretization points as follows. For each point $x_i \in X_{N_1^{(k)}}$, such that the condition (3.8) is satisfied, we add to x_i at most:

- four points (the blue circles depicted in the top-left frame of Figure 1), thus generating the set $X_{N_1^{(k+1)}}$;
- eight points (the red squares shown in the top-right frame of Figure 1), thus creating the set $X_{N_2^{(k+1)}}$.

However, when the point x_i is on or close to the boundary, the number of involved points is obviously reduced, as shown in Figure 1 (bottom). In both cases the new points are given by properly either adding or subtracting the value of (3.9) to the components of x_i. Moreover, we remark that in the illustrative example of Figure 1 the point x_i is marked by a black cross, while the new sets are such that $X_{N_1^{(k)}} \subset X_{N_2^{(k)}}$, for $k = 1, 2, \ldots$.

`Step 7`. The iterative procedure stops when there are no points anymore that satisfy the condition (3.8), returning the set $X_{N_2^{(k^*)}}$ (where k^* denotes the last iteration).

4. Numerical results

In this section we summarize the results derived from application of our adaptive refinement algorithm, which is implemented in MATLAB. All the results are carried out on a laptop with an Intel(R) Core(TM) i7-6500U CPU 2.50 GHz processor and 8GB RAM.

In the following we focus on two test problems of the form (2.3) defined on the domain $\Omega = [0,1]^2$. The exact solutions of such Poisson problems are

$$\begin{aligned} \text{P1}: \quad & u_1(x_1,x_2) = \frac{1}{20}\exp(4x_1)\cos(2x_1+x_2), \\ \text{P2}: \quad & u_2(x_1,x_2) = \exp(-8((x_1-0.5)^2+(x_2-0.05)^2)). \end{aligned}$$

A graphical representation of these analytic solutions is shown in Figure 2.

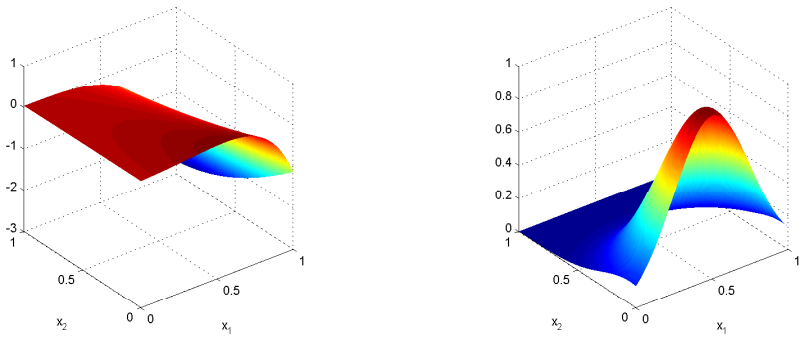

Figure 2: Graphs of exact solutions u_1 (left) and u_2 (right) of Poisson problems.

In the tests we illustrate the performance of the new adaptive scheme applied to RBF-PU collocation method by using M6 as RBF (cf. Table 1) with $\varepsilon = 3$. The two starting sets defined in Section 3 consist of $N_1{}^{(0)} = 289$ and $N_2{}^{(0)} = 1089$ grid or quasi-random Halton [6] collocation points, while the tolerances used in these experiments

are given by $tol = 10^{-5}$ and $tol = 10^{-4}$. Note that the method is basically not mesh-based, so we have no problem about possible "distortion" of the original structure of nodes. In fact, although as expected the level of achievable accuracy with non gridded points is slightly lower, the method effectively works also with scattered/random nodes. To measure the quality of our results, we give the Root Mean Square Error (RMSE), i.e.,

$$\text{RMSE} = \sqrt{\frac{1}{N_{eval}} \sum_{i=1}^{N_{eval}} |u(z_i) - \tilde{u}(z_i)|^2},$$

which is evaluated on a grid of $N_{eval} = 40 \times 40$ evaluation points. Then, in order to analyze the stability of the method, we evaluate the Condition Number (CN) of the sparse collocation matrix P in (2.7) by using the MATLAB command `condest`. As to efficiency we report the CPU times computed in seconds.

In Tables 2–3 we present the results obtained, also indicating the final number N_{fin} of discretization points required to achieve the tolerances fixed. Additionally, we report the "refined grids" after applying iteratively our adaptive algorithm. Precisely, in Figures 3–4 we plot the final collocation points obtained after applying the iterative process for test problems P1 and P2.

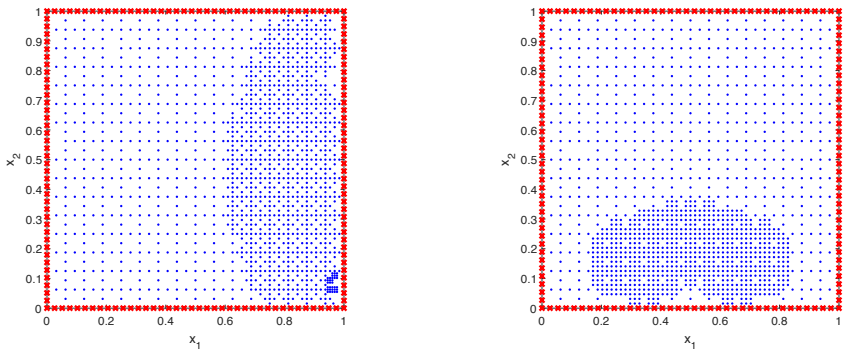

Figure 3: Final distribution of discretization points obtained by applying the adaptive algorithm with gridded points via M6, $\varepsilon = 3$ and $tol = 10^{-5}$, for problems P1 (left) and P2 (right).

Test Problem	N_{fin}	RMSE	CN	time
P1	1541	5.15e−6	4.85e+07	8.4
P2	1445	3.88e−6	1.03e+07	4.7

Table 2: Results obtained by starting from (interior) gridded points for M6 with $\varepsilon = 3$ and $tol = 10^{-5}$.

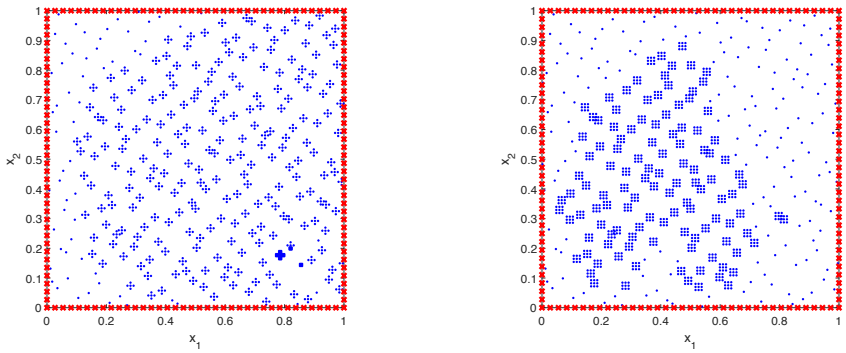

Figure 4: Final distribution of discretization points obtained by applying the adaptive algorithm with Halton points via M6, $\varepsilon = 3$ and $tol = 10^{-4}$, for problems P1 (left) and P2 (right).

Test Problem	N_{fin}	RMSE	CN	time
P1	1387	2.85e−5	3.53e+10	6.4
P2	1369	2.44e−5	4.18e+09	2.9

Table 3: Results obtained by starting from (interior) Halton points for M6 with $\varepsilon = 3$ and $tol = 10^{-4}$.

Analyzing the numerical results, we note that the adaptive algorithm increases the number of points in the regions where the solution behavior varies. As regards efficiency, then, the numerical scheme converges in few seconds in each of the tests carried out. Finally, the CN is quite similar in both cases and not so high as it often happens for traditional collocation RBF methods (see [6]).

Finally, in order to assess the advantage of our adaptive scheme, as a comparison we report the results obtained by applying the RBF-PU method on uniform grids without using adaptive refinement but with a comparable number of nodes. In particular, for problem P1 we take $N = 39^2 = 1521$ obtaining a RMSE = 3.26e−4, whereas for problem P2 we consider $N = 38^2 = 1444$ collocation points reaching a RMSE = 2.77e−5. As evident from these experiments, comparing the results with the ones shown in Table 2, the adaptive algorithm enables us to increase the precision of our numerical method.

5. Conclusions and future work

In this paper we presented an adaptive algorithm to solve time-independent elliptic PDEs such as Poisson problems. This refinement strategy is applied to the RBF-PU collocation method. In particular, we proposed an adaptive approach based on a refine-

ment technique, which consisted in comparing two approximate solutions computed on a coarser set and a finer one of collocation points. This process allowed us to detect the domain areas that needed to be refined so as to improve the accuracy of the method. Numerical results showed the performance of our algorithm on some test problems.

As future work we propose to extend our adaptive algorithm for solving other types of differential problems, including also the case of possible discontinuities.

Acknowledgments

The authors sincerely thank the anonymous referee for the insightful comments and suggestions. Moreover, they acknowledge support from the Department of Mathematics "G. Peano" of the University of Torino via 2018 project "Mathematics for applications". This work was partially supported by GNCS–INdAM 2018 project "Methods, algorithms and applications of multivariate approximation". This paper is dedicated to Prof. C. Dagnino on the occasion of her retirement.

References

[1] BUHMANN M.D., *Radial basis functions: Theory and implementation*, Cambridge University Press, Cambridge, 2003.

[2] CAVORETTO R. AND DE ROSSI A., *Adaptive Meshless Refinement Schemes for RBF-PUM Collocation*, Applied Mathematics Letters **90** (2019), 131–138.

[3] CAVORETTO R., DE ROSSI A. AND PERRACCHIONE E., *Optimal Selection of Local Approximants in RBF-PU Interpolation*, Journal of Scientific Computing **74** (2018), 1–22.

[4] DAVYDOV O. AND OANH D.T., *Adaptive Meshless Centres and RBF Stencils for Poisson Equation*, Journal of Computational Physics **230** (2011), 287–304.

[5] DRISCOLL T.A. AND HERYUDONO A.R.H., *Adaptive Residual Subsampling Methods for Radial Basis Function Interpolation and Collocation Problems*, Computers & Mathematics with Applications **53** (2007), 927–939.

[6] FASSHAUER G.E., *Meshfree approximation methods with* MATLAB, World Scientific, Singapore 2007.

[7] HERYUDONO A., LARSSON E., RAMAGE A. AND SYDOW L.V., *Preconditioning for Radial Basis Function Partition of Unity Methods*, Journal of Scientific Computing **67** (2016), 1089–1109.

[8] LARSSON E., SHCHERBAKOV V. AND HERYUDONO A., *A Least Squares Radial Basis Function Partition of Unity Method for Solving PDEs*, SIAM Journal on Scientific Computing **39** (2017), A2538–A2563.

[9] LIBRE N.A., EMDADI A., KANSA E.J., SHEKARCHI M. AND RAHIMIAN M., *A Fast Adaptive Wavelet Scheme in RBF Collocation for Nearly Singular Potential PDEs*, CMES. Computer Modeling in Engineering & Sciences **38** (2008), 263–284.

[10] OANH D.T., DAVYDOV O. AND PHU H.X., *Adaptive RBF-FD Method for Elliptic Problems with Point Singularities in 2D*, Applied Mathematics and Computation **313** (2017), 474–497.

[11] SAFDARI-VAIGHANI A., HERYUDONO A. AND LARSSON E., *A Radial Basis Function Partition of Unity Collocation Method for Convection-Diffusion Equations*, Journal of Scientific Computing **64** (2015), 341–367.

[12] WENDLAND H., *Scattered data approximation*, Cambridge University Press, Cambridge 2005.

AMS Subject Classification: 65D15, 65M70

Roberto CAVORETTO, Alessandra DE ROSSI
Department of Mathematics "G. Peano", University of Torino
Via Carlo Alberto 10, 10123 Torino, ITALY
e-mails: roberto.cavoretto@unito.it, alessandra.derossi@unito.it

Lavoro pervenuto in redazione il 16-5-19.