Antenna Design Explorer 1.0 User's Guide

Contents

1. INTRODUCTION

Antenna Design Explorer (ADE) 1.0 is a software tool with a Graphical User Interface (GUI) aiming to optimize antenna geometrical design parameters satisfying design specifications defined by the user from a given structure.

The user's inputs include: (1) an antenna performance evaluation file [in 1.0, this can be a CST Microwave Studio model or a user defined MATLAB function (e.g., MATLAB antenna toolbox-based evaluation, user defined MATLAB functions invoking other 3D EM simulators, etc.)], (2) The design variables and their ranges, (3) The optimization goal(s) and constraint(s). In the ADE 1.0 platform, four types of optimization are supported *(see Section 1.2 for more details)*. **The output** is the designed antenna layout with optimal geometrical design parameters *(see sections 4 and 5 for more details)*.

1.1 Key Features of ADE 1.0

There are a number of antenna optimizers and almost every commercial 3D EM simulation software tool has an inbuilt optimizer. **Hence, a natural question is why should one adopt ADE 1.0 or what are the advantages of ADE 1.0 compared with other optimizers?** These queries and others of a similar nature are answered as follows:

- **Efficient Global Optimization:**
  Optimizers in EM simulation tools currently available generally provide two kinds of optimization methods: (1) local optimization methods such as the Simplex Method and Sequential Quadratic Programming and (2) global optimization methods such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Local optimization methods require a good starting point, which is often not available for antenna design. Hence, using this kind of method does not guarantee a solution to the problem. Global optimization methods are shown to be very effective for antenna optimization, but they often take too much time (e.g., months) because EM simulation is time-hungry and global optimization methods need many EM simulations to find the optimum. These remain bottlenecks for most antenna optimizer users.
  In ADE 1.0, the Surrogate Model Assisted Differential Evolution Algorithm (SADEA) method [1] is included. It has been shown that the SADEA method obtains a 4 – 8 times speed improvement compared to standard global optimization methods, while getting comparable (or even better) results [1]. As such, the adoption of ADE 1.0 for your design exploration and optimization almost guarantees a high-quality design with highly reduced computational effort by decreasing 1 – 2 months' optimization time to 1 week or less.

- **Multi-objective Optimization:**
  Sometimes, antenna designers are interested in the optimal trade-off *(see Section 2 for more details)* among different specifications associated with the problem. For example, a designer may be interested in a design calling for a balance between $\max(|S_{11}|)$ and "*Gain*". In this case, a Multi-objective Optimization (MOO) approach is required *(see Section 2 for details)*. However, most of the optimizers currently available in EM design tools do not support MOO. To address this void, ADE 1.0 offers the Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) method [2]. The MOEA/D method has been shown to deliver excellent performance for antenna MOO [3].

- **Full Tutorial on Antenna Optimization including Many Useful Tips:**
  ADE 1.0 has been designed to adequately meet the usability requirements of engineers and researchers with expertise in antenna analysis and design, but without deep knowledge of optimization. In particular, there are many tips that are well known by optimization researchers that are less familiar to optimizer users. As a complementary guide for ADE 1.0, a full tutorial is provided covering many of these tips. ADE 1.0 also supports the use of these tips through its GUI. By studying the tutorials (including videos), antenna designers without much optimization expertise can quickly handle effective antenna optimization problems in a much easier and effective way. *(See Sections 4 and 5 for details.)*
  Another critical issue is that many antenna designers struggle with the setting of parameters for optimization algorithms. Though these parameters affect the performance of the optimization algorithms, more often than not, there are no clear rules on how to set them. ADE 1.0 addresses this problem by automating the parameter settings for its algorithms in instances where the user opts not to specify the parameter settings. According to the targeted problem, the appropriate parameters are adaptively calculated and set as the default within ADE 1.0 when the user opts for the automatic settings. Of course, these automatic parameters are based on real-world antenna testing. *(See section 2 for more details on parameter settings.)*

- **High Compatibility for Working With Other Tools:**
  ADE 1.0 does not attempt to duplicate the existing functions of existing EM optimizers. Rather, it is a complementary tool, which addresses their limitations by running in conjunction with them to support users. Although ADE 1.0 can be used as a standalone software tool, it is also highly compatible with most established EM Optimizers. Not only has the GUI of ADE 1.0 considered this issue, "clever" antenna design exploration methods for the effective and efficient co-usage of ADE 1.0 and optimizers available in MATLAB and/or CST are provided. (See Section 4 for details).

1.2 Optimization Problems that ADE 1.0 Solves

ADE 1.0 solves four main types of optimization problems, illustrated in Fig. 1 with a Yagi-Uda Antenna (YUA) example. These optimization problems are typical in antenna design, exploration and synthesis.

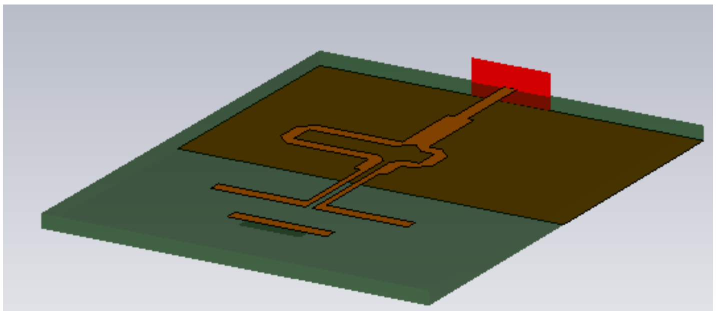


Fig. 1. 3-D Geometry of a Yagi-Uda Antenna (YUA)

- ***Specification Satisfaction***
  Example 1: The antenna designer wants to perform design exploration satisfying $\max(|S_{11}|) \le -20dB$ between 10GHz to 11GHz.

  Example 2: The antenna designer wants to perform design exploration satisfying 2 requirements between 10GHz and 11GHz:
  - $\max(|S_{11}|) \le -20dB$
  - $Gain \ge 15dB$

- ***Goal Optimization***
  Example: The antenna designer wants to perform design exploration minimizing $\max(|S_{11}|)$ between 10GHz and 11GHz.

- ***Constrained Optimization***
  Example: The antenna designer wants to perform design exploration between 10GHz and 11GHz, minimize $\max(|S_{11}|)$, while requiring $Gain \ge 15dB$.

- ***Multi-Objective Optimization***
  Example: The antenna designer wants to perform design exploration between 10GHz and 11GHz with two goals while the optimal trade-off is sought:
  - minimize $\max(|S_{11}|)$
  - maximize Gain

See Section 2 for further details on these concepts.

1.3 Downloading, Installation and Licensing

ADE 1.0 is free software and can be downloaded from the CADES ADE website. The licensing should follow the policy of Computer-aided Design and Engineering Software (CADES) Research Center (cadescenter.com) CADES Research Center is an inter-university research and development center composed of partners mainly from Europe, North America and Asia. Several key issues of licensing are:

- ADE 1.0 is prohibited from cracking, copying, distribution, retail and any other commercial activity.
- For commercial use, the interested parties or organizations must contact the CADES Research Center to obtain the necessary permits (contacting CADES is available in CADES homepage and the product pages).
- Each license expires in 90 days, but the user can download (updated) software from the CADES website (cadescenter.com) for free. This policy is to make sure that the user can keep up with the most recent version of ADE 1.0 while providing feedback for evaluation and improvement.

ADE 1.0 runs in the MATLAB environment and is compatible with Windows and Linux (not all the versions) systems. The MATLAB 2014 version or above is needed. The necessary toolboxes include the Optimization Toolbox, Machine Learning and Statistics Toolbox and Parallel Computing Toolbox. In ADE 1.0, a user-friendly seamless link with CST Microwave Studio is provided, which means ADE 1.0 can directly work with a CST model. However, CST Microwave Studio is not itself included in ADE 1.0. To use this function, the CST Microwave Studio version 2014 or above will be required. Though other 3D EM simulators are also supported on the ADE 1.0 platform, the users need to develop their own respective interfacing methods.

ADE 1.0 does not need special installation. A quick installation guide is provided in Appendix 1.

2. OPTIMIZATION CONCEPTS

This section introduces the basic concepts of optimization to the novice user. Users with existing expertise in optimization may jump to Section 3.

2.1 Design Variables

As the name suggests, design variables are parametric features in RF and EM engineering design whose metric values can be varied to improve the overall profile and quality of the design. With respect to antennas, filters and other RF or EM problems, design variables are usually measurable geometric or spatial components of the design. For instance, Fig.2 and Table 1 reveal the set design variables of a quadrifilar helical antenna (QHA) problem proposed for optimization [4].

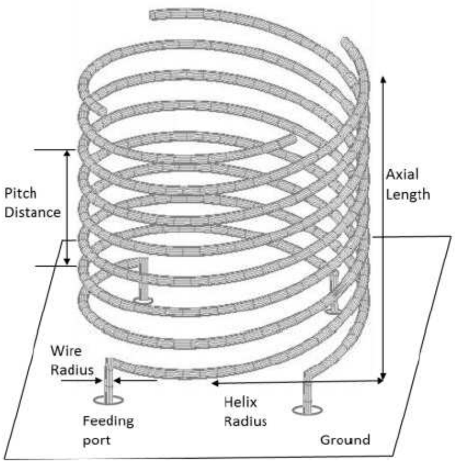


Fig. 2: 3-D Geometry of a Quadrifilar Helical Antenna (QHA) [4]

Table 1: Design Variables of a Quadrifilar Helical Antenna (QHA) [4]

QUADRIFILAR HELICAL ANTENNA (QHA) DESIGN VARIABLES		
	Minimum Value (m)	Maximum Value (m)
Pitch Distance	0.0033	0.0721
Axial Length	0.045	0.10
Radius of Helix (Top)	0.01	0.035
Radius of Helix (Bottom)	0.01	0.035
Linear Quadrature Distance Between Helical Elements	0.035	0.07
Fixed Radius of Wires	0.0005	0.0005
Fixed Distance above ground	0.01	0.01

Combining Fig. 2 and Table 1, it can be seen that design variables constitute the parametric boundaries of the QHA being examined.

2.2 Range of Design Variables

Extending the concept of design variables mentioned earlier, the range of variables for an optimization problem can be simply defined as the lower and upper bounds specified for the design variables. In the context of antennas, filters and other RF or EM problems, the range of variables are usually the minima and maxima for the measurable geometrical or spatial components of the design. Table 1 shows the range of variables (minima and maxima) for the parametric features of a quadrifilar helical antenna (QHA) proposed for optimization [4]. We generally assume that the optimal solution is not located outside of any of the above ranges.

2.3 Objective Function

In simple terms, an objective function is a mathematical equation that can model the objective(s) of an optimization problem while incorporating any constraint(s) [5]. Generally, objective functions depict optimization problems as minimization or maximization tasks. Equation 1, for example, depicts an objective function $f(x)$ for an optimization problem, "s.t" or 'subject to' constraints $g_1(x), g_2(x), g_3(x)$ and $g_4(x)$ [6]:

$$\begin{aligned}
&\min f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7 - 10x_6 - 8x_7 \\
&s.t. \\
&\quad g_1(x) = 2x_1^3 + 3x_2^4 + x_3 + 4x_4^3 + 5x_5 - 127 \le 0 \\
&\quad g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0 \\
&\quad g_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0 \\
&\quad g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0 \\
&For: \quad -10 \le x_i \le 10, i = 1,2,...7
\end{aligned} \tag{1}$$

2.4 Constraints

In optimization, a constraint is a condition that any solution to the optimization problem must satisfy. Though constraints can be specified mathematically as equalities or inequalities, they are usually depicted as inequalities in antenna design optimization [5 – 7]. For instance, the objective function in equation 1 is subject to the constraints $g_1(x), g_2(x), g_3(x)$ and $g_4(x)$.

$$\begin{aligned}
g_1(x) &= 2x_1^3 + 3x_2^4 + x_3 + 4x_4^3 + 5x_5 - 127 \le 0 \\
g_2(x) &= 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0 \\
g_3(x) &= 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0 \\
g_4(x) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0
\end{aligned} \tag{2}$$

By definition, an optimization problem subject to constraints can be called 'constrained optimization' and an optimization problem without constraints 'unconstrained optimization'.

2.5 Single Objective Optimization

From [5], an unconstrained single objective optimization problem can be described by equation 3:

$$\begin{aligned}&\min f(x)\\&s.t.\\&x\in[a,b]^d\end{aligned} \tag{3}$$

where x is the vector of decision variables; d is the dimension of x; $[a,b]^d$ are the search ranges of the decision variable x, and $f(x)$ is the objective function. The optimal solution is the value of $x\in[a,b]^d$ such that there is no other point $x*\in[a,b]^d$ with $f(x*)<f(x')$ (assuming a minimization problem).

2.6 Multi-objective Optimization

Also from [5], an unconstrained multi-objective optimization problem can be expressed as in equation 4:

$$\begin{aligned}&\min\{f_1(x),f_2(x),\ldots,f_m(x)\}\\&s.t.\\&x\in[a,b]^d\end{aligned} \tag{4}$$

where x is the vector of decision variables; d is the dimension of x; $[a,b]^d$ are the search ranges of the decision variable x, and $f_1(x)$, $f_2(x)$, …, $f_m(x)$ are the objectives of the optimization. Depending on the scope of the optimization problem and process, multi-objective optimization approaches can be broadly grouped into two classes: priori methods and posterior methods. For priori methods, a decision-maker defines preferences for the objectives and translates the multi-objective problem into a single-objective task by adopting a weighted-sum approach. In the case of posterior methods, optimal trade-off candidate solutions are presented to the decision-maker for selection. A Pareto optimal solution is a candidate solution that attains the best trade-off [5, 7].

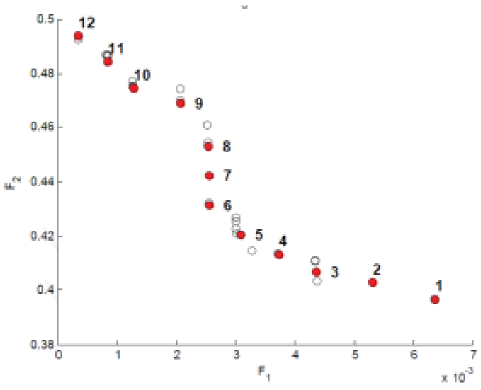


Fig. 3: Depiction of Pareto Set (PS) and Pareto Front (PF) for a Multi Objective Optimization Problem [8]

From Fig. 3, it can be seen that many (possibly, an infinite number of) Pareto optimal solutions to a multi-objective optimization are possible. Assuming x and x' are two unique solutions to (4) with $m = 2$. x is said to *dominate* x', if and only if $f_1(x) \leq f_1(x'), f_2(x) \leq f_2(x')$, and at least one of the two inequalities is strict. A candidate solution x^* is Pareto-optimal if no other candidate solution dominates it. The Pareto set (PS) is the set of all Pareto-optimal solutions and the associated image in the objective space is the Pareto front (PF) [5, 7].

2.7 Constrained Optimization

A constrained optimization problem can be depicted using equation 5 as follows:

$$\begin{aligned} &\min f(x) \\ &s.t. \\ &g_i(x) \leq 0, i = 1, 2, ..., k. \\ &x \in [a, b]^d \end{aligned} \qquad (5)$$

where $f(x)$ depicts the optimization goal and $g_i(x)$ are the constraints. Constrained optimization can be single-objective or multi-objective. In antenna optimization, we generally consider that feasible solutions (satisfying all the constraints) are better than infeasible solutions. For two feasible solutions, their optimality is determined by the objective function value or Pareto dominance.

2.8 Penalty Functions

Often described in the context of constrained optimization, penalty functions are given to penalize solutions violating the constraints. Penalty functions associate constraints with (an) objective function(s) to construct new objective function(s), making the satisfaction of the constraints self-included when optimizing the new objective function(s) [9]. Take equation 1 as an example. $f(x)$ in equation 1 is the objective function, and $g_1(x), g_2(x), g_3(x)$ and $g_4(x)$ are the constraints (inequalities) associated with the objective function. The constrained problem $\min f(x)$ can be solved as an unconstrained minimization problem as follows:

$$\min f'(x) = f(x) + \phi_i \sum_{i=1}^{4} d(g_i(x)) \qquad (6)$$

$\min f'(x)$ represents the penalized function of $f(x)$ in equation 1. For $i = 1, 2, 3, 4$, $d(g_i(x))$ is the penalty function and ϕ_i is the penalty coefficient. Equation 6 can be further expanded as follows:

$$\begin{aligned} &\min f'(x) = \min f(x) + [\phi_1 \max(g_1(x), 0)] + [\phi_2 \max(g_2(x), 0)] + ... \\ &[\phi_3 \max(g_3(x), 0)] + [\phi_4 \max(g_4(x), 0)] \end{aligned} \qquad (7)$$

3. OPTIMIZATION METHODS IN ADE 1.0

This section introduces the three optimizers in ADE 1.0 and their respective parameter settings. The algorithmic details of the three selected optimizers are *not* covered. Instead, their properties and the recommended conditions to use them are provided. Readers who are interested in further details on the optimizers can refer to the following papers: [1 – 5].

3.1 The Surrogate-Assisted Differential Evolution Algorithm (SADEA) Optimizer

The foundation of the SADEA algorithm is the surrogate model-assisted evolutionary search (SMAS) framework [1, 5], whose goal is to solve global optimization problems with computationally expensive evaluations (such as EM simulations). In surrogate model-assisted optimization, a computationally cheap approximation model, based on statistical learning techniques, is built to predict the response of computationally expensive evaluations, so as to improve the efficiency. The method of making surrogate modeling and evolutionary search work harmoniously (also called model management) is the crux of surrogate model-assisted optimization. SMAS shows clear advantages compared to several popular surrogate model-assisted global optimization frameworks [7]. SADEA adapts the SMAS framework to the antenna design exploration domain and some algorithm operators or parameters are determined considering landscape characteristics of antenna problems and verified by real-world antenna design problems. SADEA [1] was improved in [10], which is embedded in ADE 1.0. Another advantage of SADEA is that it is able to handle antenna design exploration problems with up to around 30 design parameters, while keeping both the optimization quality and efficiency high.

The SADEA algorithm is suitable for all antenna design problems and supports specification satisfaction (type 1), goal optimization (type 2) and constrained optimization (type 3) *(refer to Section 1 for more details)*. SADEA often needs no more than a few hundred EM simulations to find the optimum for antenna problems. SADEA is mostly suitable for problems where EM simulation is time costly.

3.2 The MOEA/D-DE Optimizer

The MOEA/D framework [2] is a state-of-the-art MOO method. The principle is to decompose a multi-objective optimization problem into a number of single objective optimization sub-problems and evolve simultaneously utilizing neighborhood relations to achieve the approximate Pareto front (PF). The MOEA/D framework shows clear advantages compared to several popular multi-objective optimization methods, such as the widely used NSGA-II and SPEA2. The MOEA/D-DE algorithm is a successful example based on the MOEA/D framework. [2] demonstrates its high performance in antenna design exploration. Consequently, the MOEA/D-DE algorithm is embedded in ADE 1.0. Note that the MOEA/D-DE algorithm does not handle efficient optimization. Hence, to avoid optimization time of unacceptable duration, the fidelity of the EM simulation model should be low enough and the number of design variables should not be large.

The MOEA/D-DE algorithm is suitable for all antenna design problems and supports multi-objective optimization (type 4) *(refer to Section 1 for details)*. The MOEA/D-DE algorithm often needs thousands of EM simulations to find an approximate Pareto front (PF) for antenna problems.

3.3 The DE Optimizer

Differential Evolution (DE) [11] is a standard global optimization method. Its excellent performance for antenna design problems is shown in [1, 10]. It is generally recognized as being better than Genetic Algorithms (GAs); comparable to Particle Swarm Optimization (PSO), and worse than the Covariance Matrix Adaptation Evolution Strategy. These three algorithms have been embedded in the optimizers of several EM simulation tools, such as CST and FEKO. Therefore, DE is embedded in ADE 1.0 to compensate for existing tools. As for SADEA and MOEA/D-DE, the DE algorithmic parameters are determined based on real-world antenna optimization tests. As with MOEA/D-DE, DE cannot address efficient optimization, and its optimization capability is comparable to SADEA.

The DE algorithm is suitable for all antenna design problems and supports type 1, type 2 and type 3 optimization (see Section 1). The DE algorithm often needs thousands of EM simulations to find the optimum. Although SADEA can replace DE (comparable optimization quality but much faster) for most antenna design exploration problems, DE is suited to problems with very cheap evaluations (such as analytical functions). For this kind of problem, optimization efficiency is no longer a problem.

Table 2 summarizes the recommended conditions for selecting the optimizers available in ADE 1.0.

Table 2: Recommended Antenna Design Exploration Problems for ADE 1.0 Optimizers

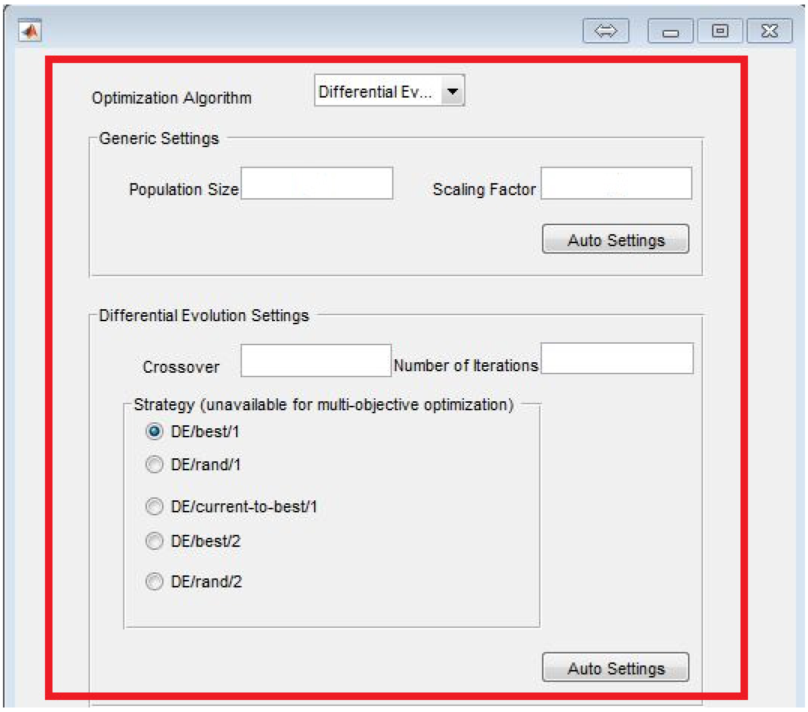
Optimizer	Problem Type	Recommended Number of Design Variables	Evaluation Time per Candidate Design
SADEA	1, 2 and3	Up to 40	Long (ideally less than 30 minutes / simulation)
MOEA/D	4	Less than or around 20	Short
DE	1, 2 and 3	Up to 60	Short

Besides these three optimizers, it is also worth noting that ADE 1.0 permits users to employ customized algorithms or evaluation functions for antenna design exploration.

3.4 Parameter Settings for Optimizers

Setting algorithmic parameters is often a challenge for antenna designers not familiar with optimization methods. To address this, ADE 1.0 offers an automatic setting for all its built-in optimization methods; alternatively, experts in optimization methods can instead specify values for algorithmic parameters. The automatic setting feature in ADE 1.0 uses calculations based on a set of formulas derived from real-world antenna optimization experience.

- ***Parameter Settings for Differential Evolution (DE)***
  Referring to the ADE 1.0 window in Fig. 5, the following settings are recommended for the user to adopt:



Optimization Algorithm Differential Ev...

Strategy (unavailable for multi-objective optimization)

● DE/best/1

○ DE/rand/1

○ DE/current-to-best/1

○ DE/best/2

○ DE/rand/2

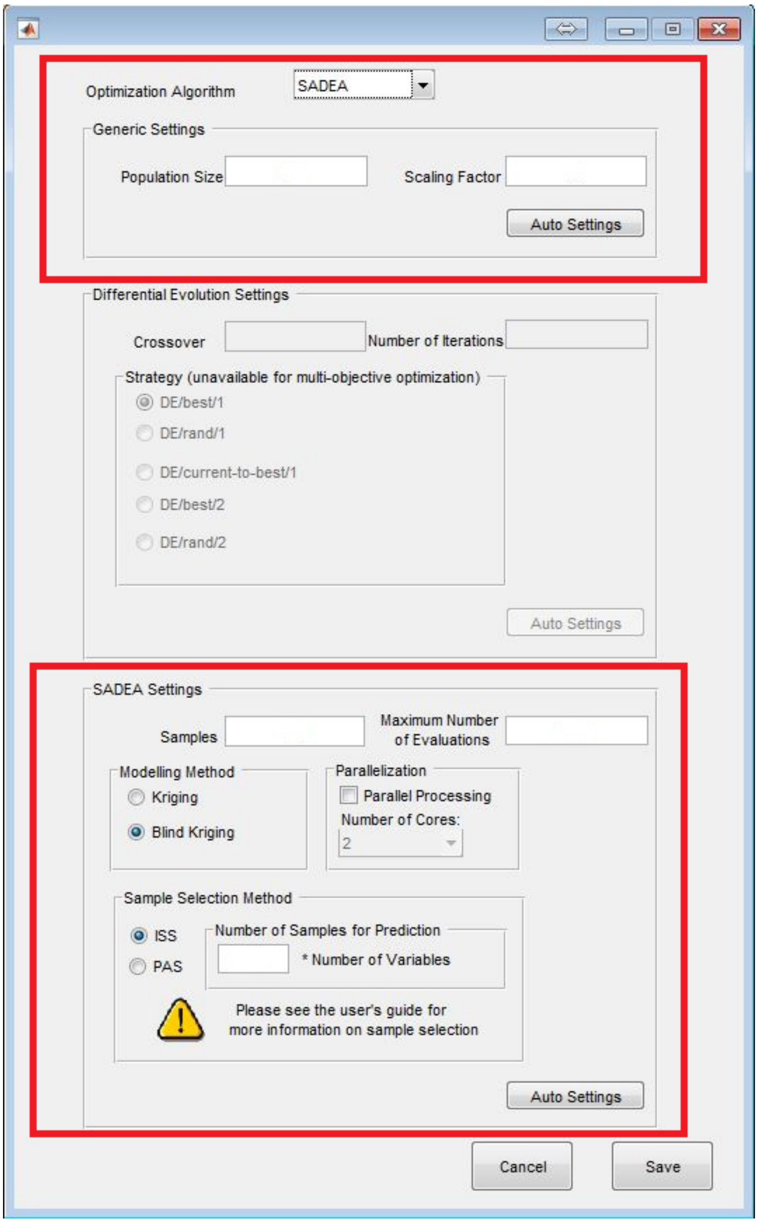
Fig. 5: ADE 1.0 Window for Differential Evolution (DE) Parameter Settings

- ***Population Size:*** The population size should be between 40 and 60 for 10–30–dimensional problems.
- ***Scaling Factor:*** As a rule of thumb, a scaling factor of 0.8 – 1.0 is suggested.
- ***Crossover Rate:*** For antenna design exploration, a crossover rate between 0.7 and 0.9 is recommended. With respect to convergence, a crossover rate specified in the upper bound region (near 0.9) may yield faster convergence.
- ***Number of Iterations:*** The choice of the number of iterations is problem specific. However, ADE 1.0 is designed to graphically display the progress of the optimization using plots of convergence trend and population diversity. These plots can inform the user whether the number of iterations specified is adequate. *(See section 4 for more details.)*
- ***Strategy Specification:*** The DE algorithm of ADE 1.0 offers the user a choice from the five common crossover methods available for DE. Depending on the problem description and as depicted in the ADE 1.0 window, the user can choose any one of the following methods or strategies:
    - ***DE/best/1:*** Popular for antenna problems.
    - ***DE/rand/1:*** Popular for antenna problems.
    - ***DE/current-to-best/1:*** Highly recommended considering both solution quality and efficiency (default)
    - ***DE/best/2:*** More suitable for exceptional cases requiring large diversities. Note that convergence may be slow.

- ***DE/rand/2:*** More suitable for exceptional cases requiring even larger diversities. Note that convergence may be slow.

- ***Parameter Settings for Surrogate-Assisted Differential Evolution Algorithm (SADEA)***

  Referring to the ADE 1.0 window in Fig. 6, the following settings are recommended for the user to adopt:



Optimization Algorithm: SADEA

Generic Settings

Population Size:

Scaling Factor:

Differential Evolution Settings

Crossover:

Number of Iterations:

Strategy (unavailable for multi-objective optimization)

● DE/best/1

○ DE/rand/1

○ DE/current-to-best/1

○ DE/best/2

○ DE/rand/2

SADEA Settings

Samples:

Maximum Number of Evaluations:

Modelling Method

○ Kriging

● Blind Kriging

Parallelization

□ Parallel Processing

Number of Cores: 2

Sample Selection Method

● ISS

○ PAS

Number of Samples for Prediction: * Number of Variables

Please see the user's guide for more information on sample selection

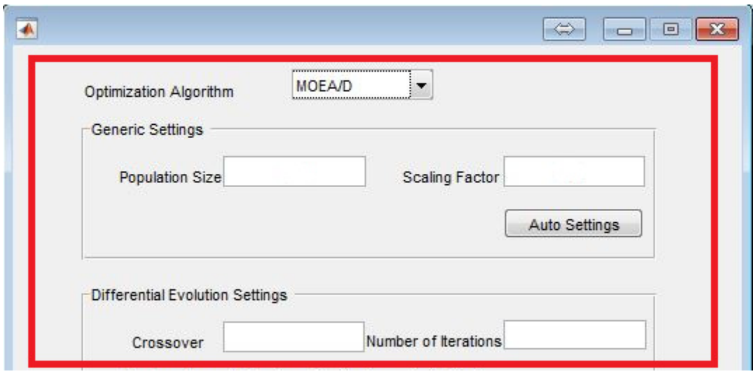
Fig. 6: ADE 1.0 Window for Surrogate Assisted Differential Evolution Algorithm (SADEA) Parameter Settings

- ***Population Size:*** The population size should be between 40 and 60 for 10–30–dimensional problems.
- ***Scaling Factor:*** As a rule of thumb, a scaling factor of 0.8–1.0 is suggested.

- ***Crossover Rate:*** For antenna design exploration using the SADEA method, the crossover rate is self-adaptive.
- ***Samples:*** It is highly recommended that the number of initial samples is set between 5xD and 10xD (D is the number of design variables). It is worth noting that 5xD is often enough.
- ***Number of Iterations:*** The choice of the number of iterations is problem specific. However, ADE 1.0 is designed to graphically display the progress of the optimization using plots of convergence trend and population diversity. These plots can inform the user whether the number of iterations specified is adequate.
- ***Modelling Method:*** The SADEA method featured in ADE 1.0 offers two modeling methods: *Kriging* and *Blind Kriging*. The Blind Kriging modeling method is two times slower than the Kriging modeling method but could be a better choice for antenna problems with larger dimensions (around 20-30 dimensions).
- ***Sample Selection Method:*** The SADEA method in ADE 1.0 offers two sampling selection methods: *Promising Area-based training Data Selection (PAS)* and *Individual Solution-based Training Data Selection (ISS)*. For both ISS and PAS, the Number of Variables for Sample Prediction is strongly recommended to be between 5-10 x D (D is the number of design variables). It is worth noting that more samples within this range (5-10 x D) may yield better prediction quality. Generally, ISS works for almost all antenna problems but takes more time than PAS. On the other hand, PAS works for most antenna problems expect those with a narrow optimal region (to the best of our knowledge, only dielectric antennas). Note that the training time grows exponentially with the dimensionality. Hence, for problems with less than 10 variables, the training times of PAS and ISS do not show a significant difference. For problems in more than 20 dimensions, the training time differs considerably. For such problems, if ISS is unavoidable, then the *kriging* modeling method can be used to reduce the training time.

- ***Parameter Settings for Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D)***

  Referring to the ADE 1.0 window in Fig. 7, the following settings are highly recommended for the user to adopt:



Optimization Algorithm: MOEA/D

Fig. 7: ADE 1.0 Window for Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D) Parameter Settings

- ***Population Size:*** For most antenna problems, the population size is recommended to be between 100 and 200, relative to the dimension of the problem being evaluated.
- ***Scaling Factor:*** We recommend setting the scaling factor between 0.4 and 0.6.
- ***Crossover Rate:*** For antenna design exploration using the MOEA/D method, we recommend setting the crossover rate between 0.9 and 1.0.
- *For the MOEA/D method, all other parameters adopted in ADE 1.0 are self-adaptive.*

- **Penalty Coefficients Settings**

For constrained optimization problems, the penalty coefficient must be specified as shown in the ADE 1.0 window of Fig. 8:

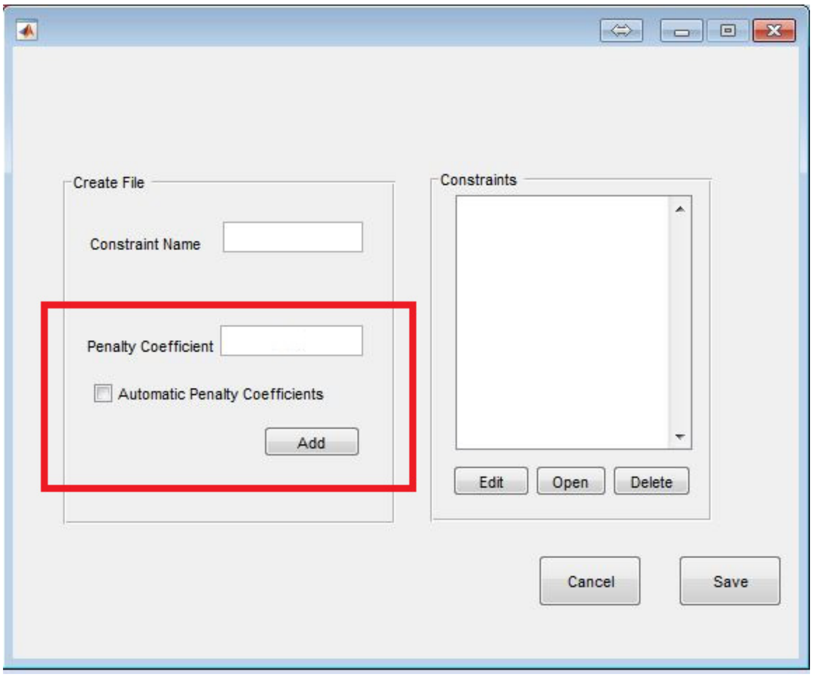


Fig. 8: ADE 1.0 Window for Penalty Coefficients' Settings

From Fig. 8, it can be seen that ADE 1.0 allows the user to automate settings for penalty coefficients to be adopted in the constraint function such that they augment the objective function. However, for manual input, it is strongly advised that the penalty coefficient be large enough to clearly discriminate the feasible and infeasible solutions.

4. BASIC USE OF ADE 1.0

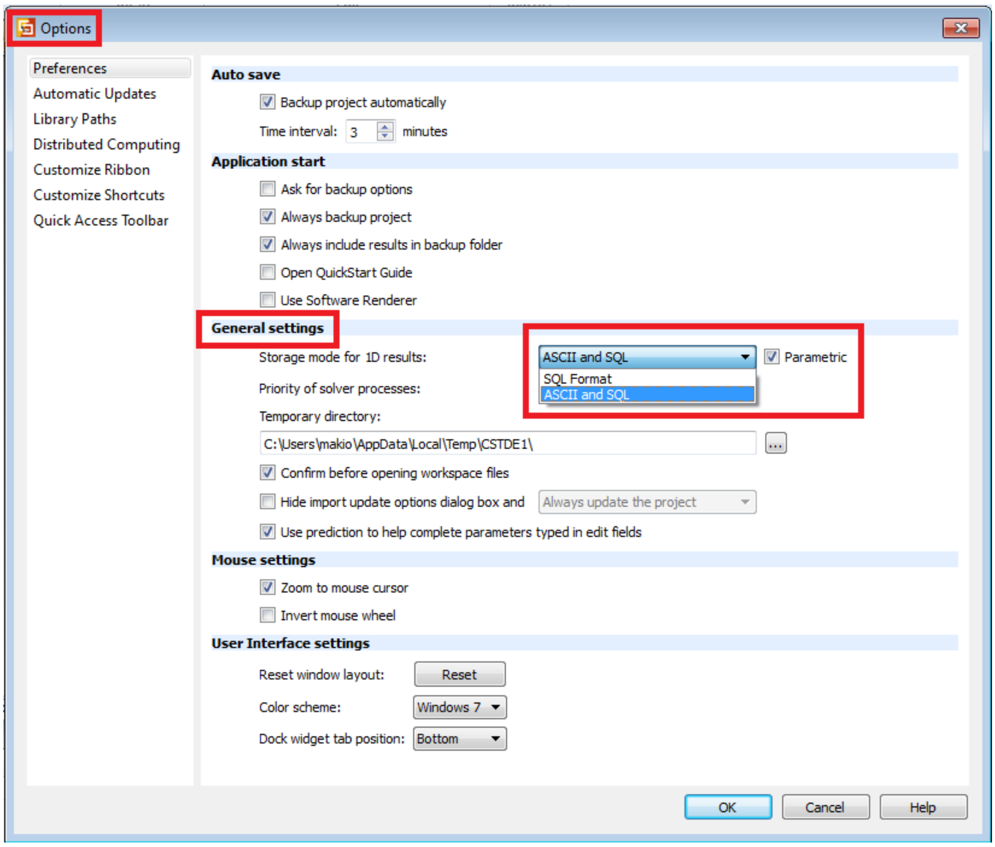
This section provides the basic instructions on how to use ADE 1.0. ADE 1.0 is built in MATLAB. Hence, all the MATLAB functions can be used easily with the problems set in ADE 1.0. ADE 1.0 also provides the packed objective function (including the interfacing between MATLAB and CST Microwave Studio), which can be used in any function or script as long as it is an m-file. ***It is strongly recommended for the user to watch the tutorial videos and engage this instruction set as a COMPLEMENTARY TOOL ONLY.***

4.1 Using the Software

ADE 1.0 offers a dynamic and easy-to-use GUI. First-time users are strongly encouraged to watch the tutorial videos.

- ***Run the Software***

  Before executing ADE 1.0, the user needs to check and confirm the data format for CST outputs or results. ***To avoid computational errors due to missing CST outputs or results, the storage mode for CST outputs or results must be set to ASCII and SQL.*** As shown in Fig. 9, this can be achieved by changing the general settings in the file options of CST Microwave Studio.



Auto save

☑ Backup project automatically

Time interval: 3 minutes

Application start

☐ Ask for backup options

☑ Always backup project

☑ Always include results in backup folder

☐ Open QuickStart Guide

☐ Use Software Renderer

General settings

Storage mode for 1D results: ASCII and SQL ☑ Parametric

SQL Format

ASCII and SQL

Priority of solver processes:

Temporary directory:

C:\Users\makio\AppData\Local\Temp\CSTDE1\

☑ Confirm before opening workspace files

☐ Hide import update options dialog box and Always update the project

☑ Use prediction to help complete parameters typed in edit fields

Mouse settings

☑ Zoom to mouse cursor

☐ Invert mouse wheel

User Interface settings

Reset window layout: Reset

Color scheme: Windows 7

Dock widget tab position: Bottom

Fig. 9: CST Output Settings for ADE 1.0

- ADE 1.0 can be executed in the MATLAB environment by adding the software folder and subfolders to MATLAB's current working path.
- To launch the software, type/enter ***CADES_EM_Run*** in the MATLAB command window.

- ***Creating a New Project or Opening an Existing Project***

  - A new project is created and named or an existing project opened by clicking the appropriate icon, as shown in Fig. 10.

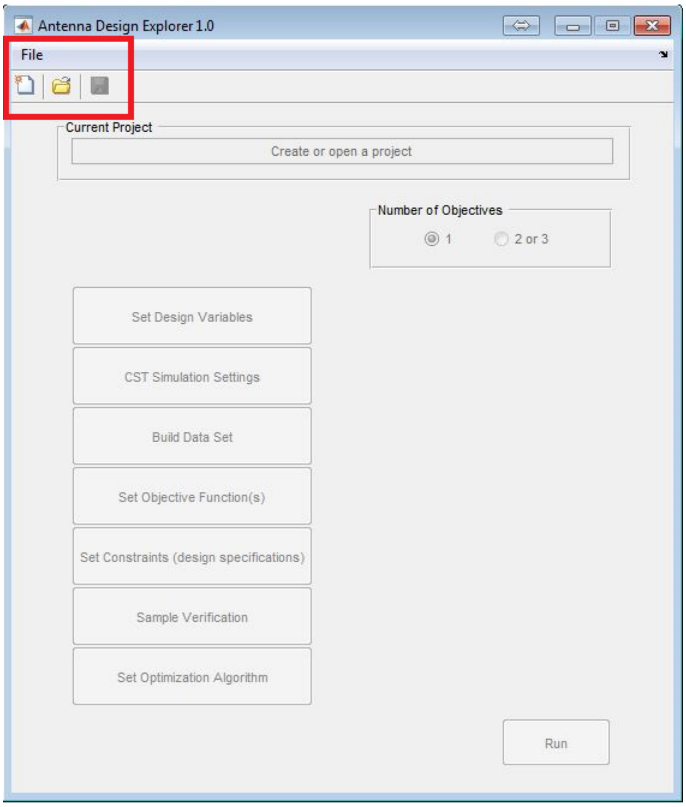
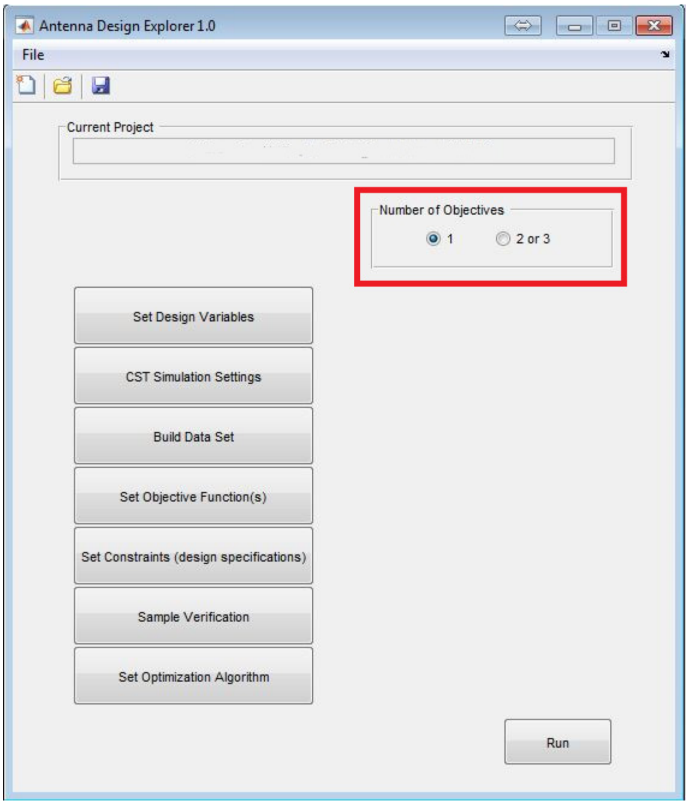


Fig. 10: Creating or Opening a Project File in ADE 1.0

- ***Selecting the Number of Objectives***

  - As shown in Fig. 11, the number of objectives is selected for the optimization problem: single or multiple.



Number of Objectives

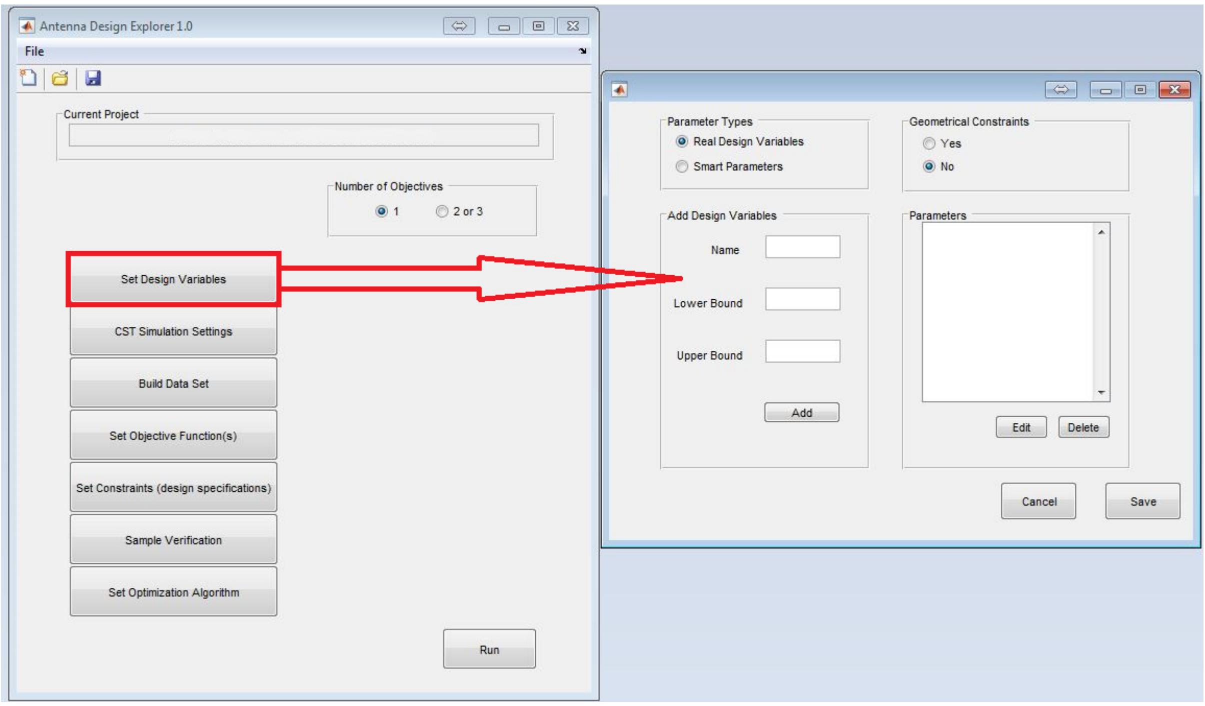
● 1 ○ 2 or 3

Antenna Design Explorer 1.0
File
Current Project
Set Design Variables
CST Simulation Settings
Build Data Set
Set Objective Function(s)
Set Constraints (design specifications)
Sample Verification
Set Optimization Algorithm
Run

Fig. 11: Setting the Number of Objectives for ADE 1.0

- ***Setting the Design Variables***

  - To set the design variables (for real or smart parameters), the lower and upper bounds are specified accordingly, as shown in Fig. 12.
  - ***For optimization problems involving CST models, it is strongly recommended that the user double-checks the parameter list of the CST model to ensure consistency with the variable names being specified in ADE 1.0.***
  - In the case of engaging smart parameters for some optimization problems, the real design parameters must be derived from the smart parameters using a function file, automatically generated by ADE 1.0.
  - *(See the video tutorials for more details "ADE 1.0".)*



Number of Objectives: ● 1 ○ 2 or 3

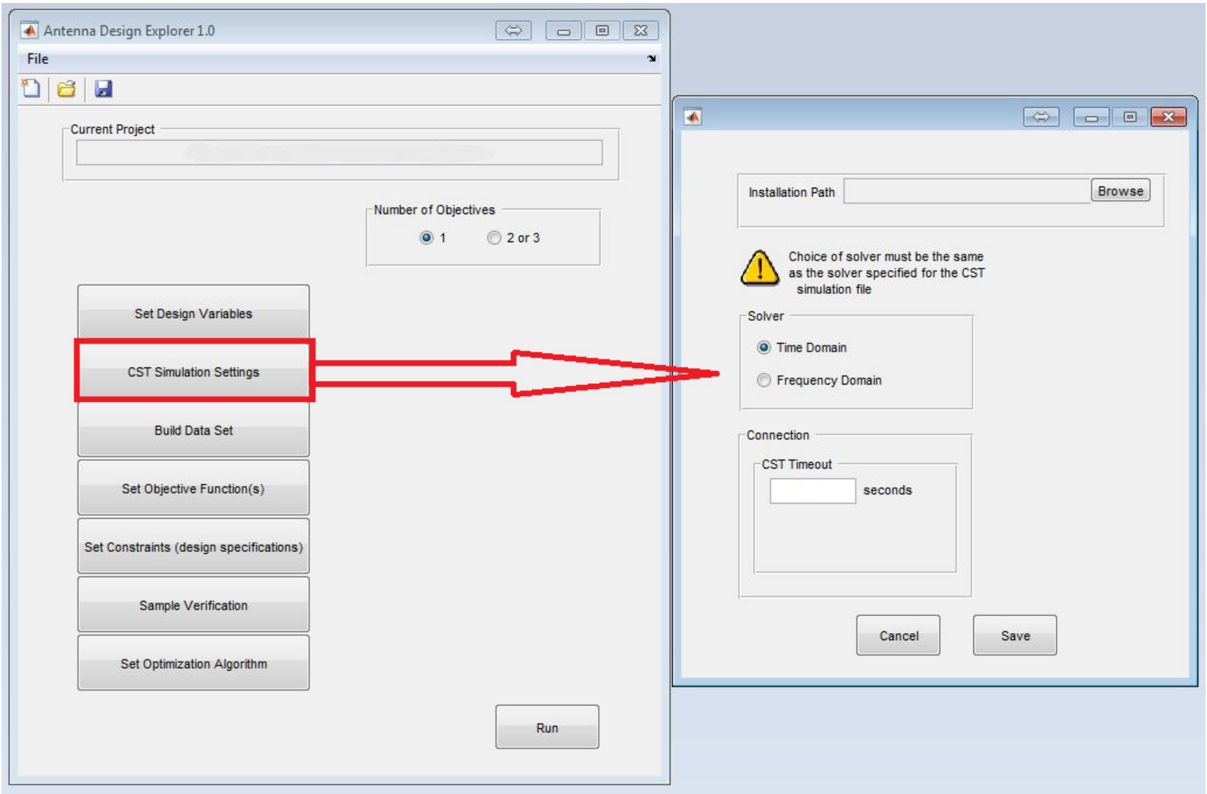
Parameter Types: ● Real Design Variables ○ Smart Parameters

Geometrical Constraints: ○ Yes ● No

Fig. 12: Adding and Editing the Design Variables in ADE 1.0

- ***CST Simulation Settings***

    - As shown in Fig. 13, CST simulation settings can be set by selecting and specifying the installation path (CST Design Environment), the solver (Time Domain or Frequency Domain) and the CST Timeout. ***(The CST Timeout is recommended to be 2 - 3 times the approximate duration in seconds of the simulation time for the antenna model in CST Microwave Studio.)***
    - Note that, if the wrong timeout duration is set, the CST Microwave Studio parallel simulation pool will timeout during ADE 1.0 runtime, hampering the optimization process. This will be repeatedly reported in an onscreen error message.
    - *(See the video tutorials for more details "ADE 1.0".)*



Choice of solver must be the same as the solver specified for the CST simulation file

Fig. 13: CST Simulation Settings for ADE 1.0

- ***Build Data Set***

  - The data set is built by specifying the response name and selecting the CST model for the antenna to be optimized. As depicted in Fig. 14, a CST simulation is initiated seamlessly from ADE 1.0 to generate response signals unique to the selected CST model.
  - Upon the completion of the simulation, the response signal of interest is selected and used in the objective function and constraints.
  - If not using CST Microwave Studio "Other Methods (function file)" may be selected.
  - *(See the video tutorials for more details "ADE 1.0".)*

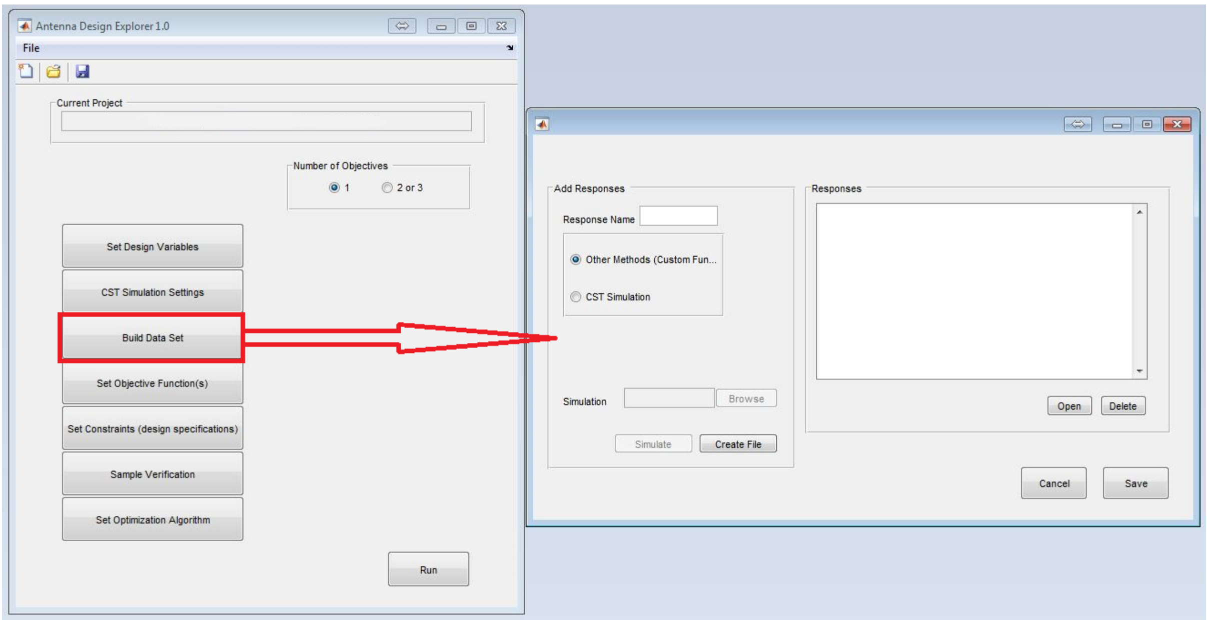
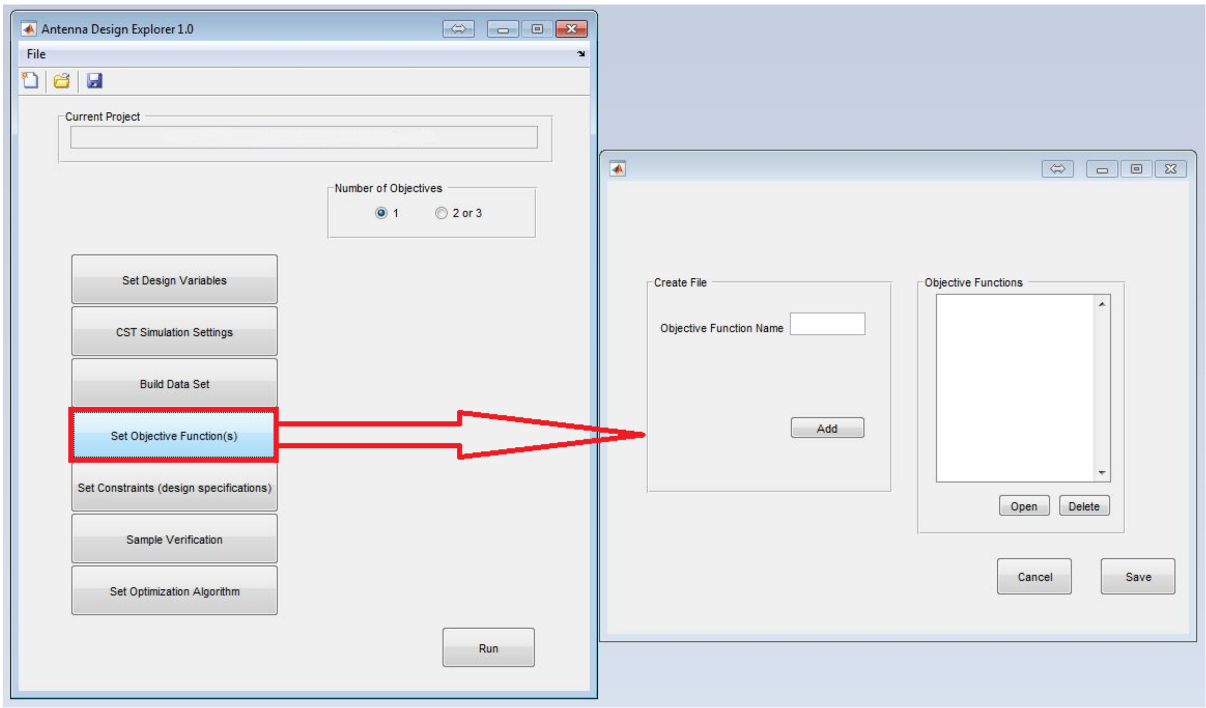


Fig. 14: Building the Data Set for ADE 1.0

- ***Set Objective Function(s)***

  - From Fig. 15, the objective function resulting from the selected response signal is named and set by updating an automatically generated generic MATLAB function (as detailed in the function files note). *(See section 4.2.)*
  - *(See the video tutorials for more details "ADE 1.0".)*



Antenna Design Explorer 1.0

Current Project

Number of Objectives

● 1 ○ 2 or 3

Fig. 15: Setting the Objective Function (s) for ADE 1.0

- ***Set Constraints (Design Specifications)***

    - Constraints, if any, can be defined and specified as required. The constraint name and associated penalty coefficient are to be specified. From Fig. 16, a generic MATLAB function file is generated, with a name that corresponds to the name specified for the constraint. The user can update the function to suit the constrained optimization problem.
    - *(See the video tutorials for more details "ADE 1.0".)*
    - *(Refer to section 2 for more details in penalty coefficient setting.)*

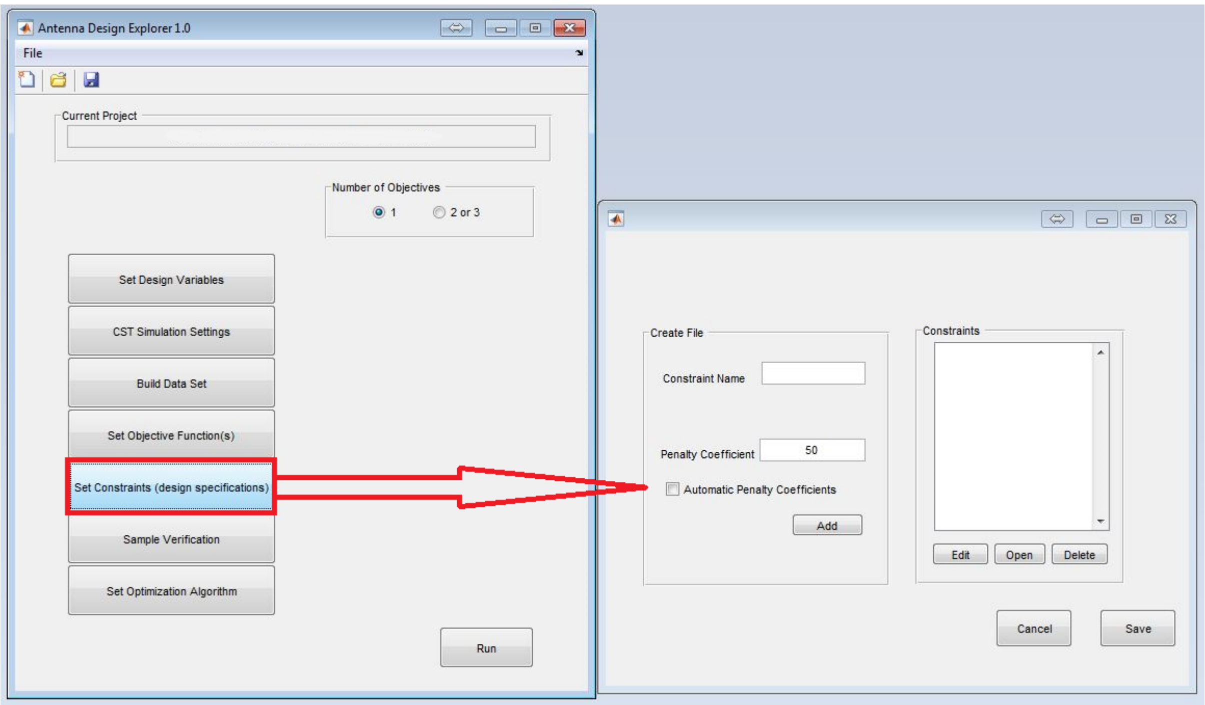
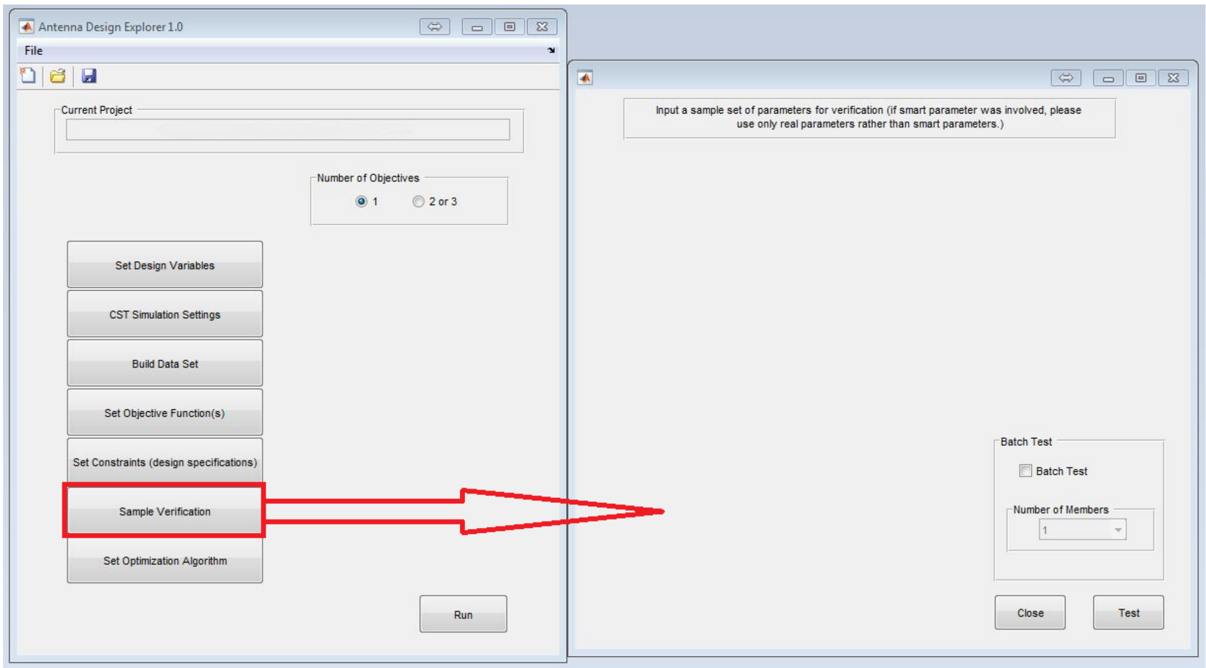


Fig. 16: Setting the Design Constraints in ADE 1.0

- ***Sample Verification***

  - Sample verification is carried out using a random set of variables within the specified bounds (lower and upper) of the design variables, either by entering the variables manually or through a “Batch Test”, as shown in Fig. 17.
  - ***Note that, for the initial sample verification, the simulation process is carried out twice, to confirm that the defined inputs match the output. Upon confirmation, the simulation process runs only once for the subsequent sample verification.***
  - *(See the video tutorials for more details “ADE 1.0”.)*


Antenna Design Explorer 1.0
File
Current Project
Set Design Variables
CST Simulation Settings
Build Data Set
Set Objective Function(s)
Set Constraints (design specifications)
Sample Verification
Set Optimization Algorithm
Run
Batch Test
Close
Test

Number of Objectives: ● 1 ○ 2 or 3

Input a sample set of parameters for verification (if smart parameter was involved, please use only real parameters rather than smart parameters.)

Batch Test: □ Batch Test

Number of Members: 1

Fig. 17: Sample Verification in ADE 1.0

- ***Setting the Optimization Algorithm***

  - Depending on the problem-specific optimization the user requires, ADE 1.0 to perform, any one of the three built-in algorithms in ADE 1.0 can be selected. *(See section 3 for fuller details.)*

4.2 Essential Tutorials

- ***CST Response Signals: Interpretation and Translation***

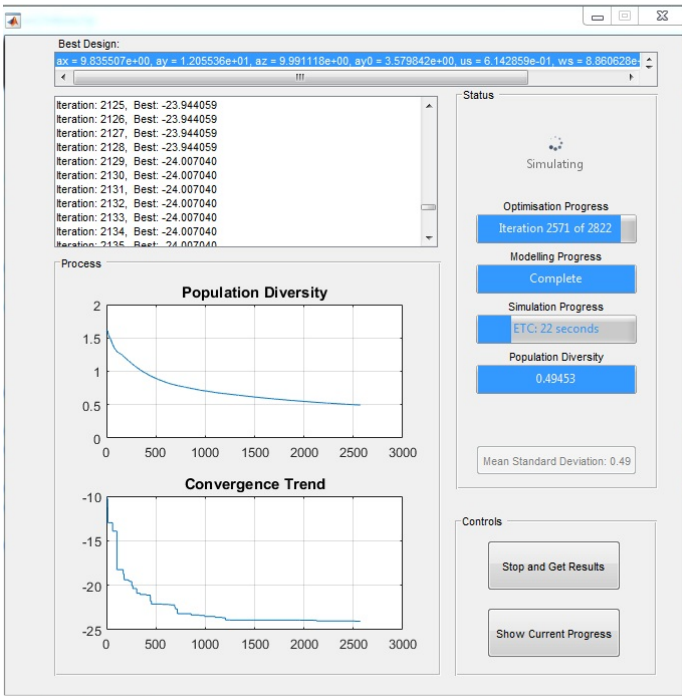
  - Table 3 provides a list of response signals and their respective interpretations and translations, based on CST 2014-2016. *Please note that the table is to serve as a guide only.* ***It is strongly recommended for users to always check to ensure the correct response signal output from CST is selected for the optimization process.***

Table 3: CST Response Signals (Interpretation and Translation)

RESPONSE SIGNALS	CST TO MATLAB (ADE 1.0) TRANSLATION	INTERPRETATION
Gain	**Gain (IEEE),Theta=x,Phi=y.sig**	Gain response signal, where ‘x’ and ‘y’ correspond to the specified angles of cut through the horizontal and vertical cut planes, respectively, of the radiation pattern
S – Parameter (Magnitude)	**ai(1)j(1).sig**	Magnitude form of the S-parameter (scattering parameter) response signal, where ‘i’ and ‘j’ correspond to the output and input ports, respectively.
S – Parameter (Complex)	**cSi(1)j(1).sig**	Complex form of the S-parameter (scattering parameter) response signal, where ‘i' and ‘j’ correspond to the output and input ports, respectively.
S – Parameter (dB)	**di(1)j(1).sig**	‘dB’ form of the S-parameter (scattering parameter) response signal, where ‘i’ and ‘j’ correspond to the output and input ports, respectively.
S – Parameter (Phas)	**pi(1)j(1).sig**	Phasor form of the S-parameter (scattering parameter) response signal, where ‘i' and ‘j’ correspond to the output and input ports, respectively.
Radiation Efficiency	**signal_default.sig**	Radiation efficiency response signal.
Total Efficiency	**signal_default_lf.sig**	Total radiation efficiency response signal.

- ***Observation and Interpretation of Results***

  - In viewing and estimating results from the optimization process, it is worth noting that two important factors, showing the progress of the optimization process, are convergence trend and diversity of the population.



Best Design:

ax = 9.835507e+00, ay = 1.205536e+01, az = 9.991118e+00, ay0 = 3.579842e+00, us = 6.142859e-01, ws = 8.860628e-

```
Iteration: 2125,  Best: -23.944059
Iteration: 2126,  Best: -23.944059
Iteration: 2127,  Best: -23.944059
Iteration: 2128,  Best: -23.944059
Iteration: 2129,  Best: -24.007040
Iteration: 2130,  Best: -24.007040
Iteration: 2131,  Best: -24.007040
Iteration: 2132,  Best: -24.007040
Iteration: 2133,  Best: -24.007040
Iteration: 2134,  Best: -24.007040
Iteration: 2135,  Best: -24.007040
```

Optimisation Progress: Iteration 2571 of 2822

Modelling Progress: Complete

Simulation Progress: ETC: 22 seconds

Population Diversity: 0.49453

Mean Standard Deviation: 0.49

Fig. 18: ADE 1.0 Field Monitors for Optimization Process

  - ***Population Diversity:*** Within the population, if the standard deviation among individuals is high, then the diversity is high. In a similar fashion, if the standard deviation among individuals of the population is small, then the diversity is low. For the test case shown in Fig. 18, the optimization process has little room for improvement when the diversity is low: at a value of 0.5 and approaching zero on the diversity field monitor.

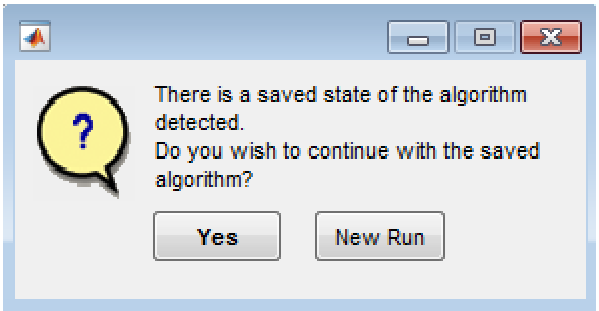
  - **Convergence Trend:** In ADE 1.0., the performance of the current best solution to the optimization problem can be easily observed on the convergence trend field monitor. For the test case given in Fig. 18, it can be seen that the result is already very good after 500 EM simulations. After about 1200 EM simulations, the convergence is quite clear and the population diversity is small. Hence, there is no need to continue.

- ***Optimization Timeout***

  - The ADE 1.0 optimization process is halted after the maximum number of iterations has been reached. If the field monitors (population diversity and convergence trend) strongly suggest that there is room for improvement, the optimization process can be continued by carrying out the following steps:

- ***Re-opening your Project File in ADE 1.0.:***

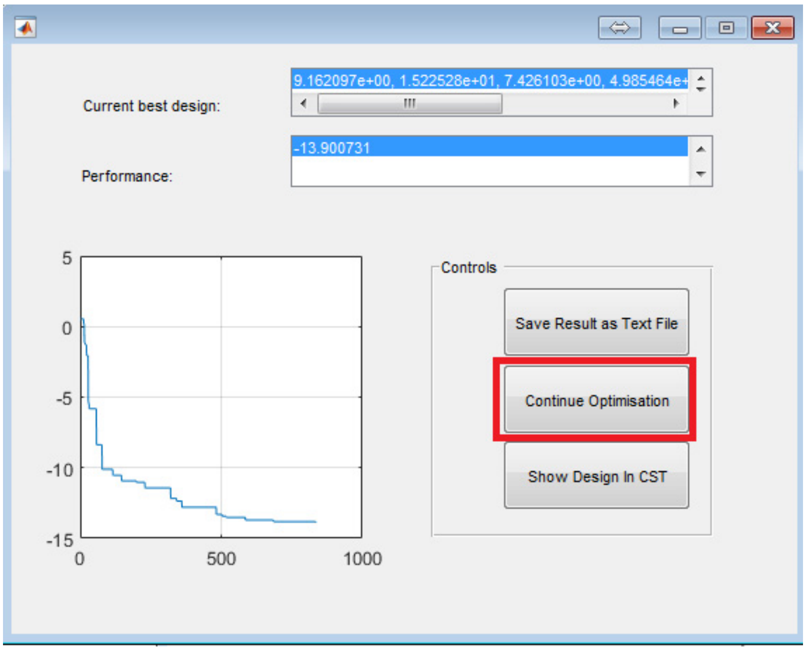
- To do this, you need to click “Open” and select the “project.mat” file corresponding to the name of your current project in the appropriate folder: all settings will be loaded. By clicking “Optimize” in ADE 1.0, a new window will pop up, asking the user whether to continue an existing run or start a new run, as shown in Fig. 19. Select and click on “Yes” as shown in Fig. 19. ***(Ignore this step if your “project.mat” is already opened.)***



There is a saved state of the algorithm detected.
Do you wish to continue with the saved algorithm?

Fig. 19: ADE 1.0 Dialogue Box for Continuing an Optimization Run

- On clicking “Yes”, a new window appears. Select and click on “Continue Optimization”, as shown in Fig. 20.



Current best design: 9.162097e+00, 1.522528e+01, 7.426103e+00, 4.985464e+

Performance: -13.900731

Fig. 20: ADE 1.0 Dialogue Box for Continuing Optimization Run (2)

- On clicking “Continue Optimization”, a new window appears. Specify the number of evaluations or iterations you require ADE 1.0 to continue running with and click “OK” to continue the optimization process, as shown in Fig. 21.

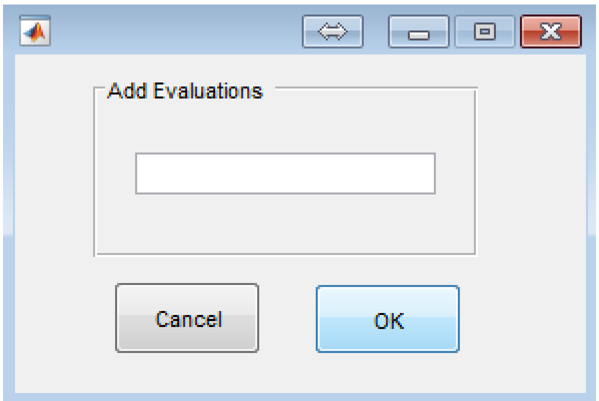


Fig. 21: ADE 1.0 Dialogue Box for Continuing an Optimization Run

- *Setting Smart Parameters: DRA Test Case*

  - From the ADE 1.0 window in Fig. 12, upon the selection of smart parameters, a MATLAB function is automatically generated for the user to update for the derivation of real value(s) from smart parameter(s). A test case has been used as an example, using the dielectric antenna (DRA) problem, in the video tutorial available through this link: CADES Research Center (2016), Dielectric Resonator Antenna (DRA) Optimization Using ADE 1.0

- *User-defined MATLAB Functions*

  - **MATLAB Function Files: Auto M-Files**

  - To evaluate smart parameters, objective functions and constraints in ADE 1.0, corresponding MATLAB function files (M-files) are generated automatically as soon as user-defined inputs are entered in the corresponding dialogue boxes and/or windows. Most of these auto M-files serve as go-betweens for CST Microwave Studio simulation and MATLAB computation processes. Generic auto M-files have to be updated by the user to specifically address the optimization problem. Depending on the intended use, the typical layout of an auto M-file within ADE 1.0 allows the user to specify the design variable(s) and/or response signal(s) from CST MW studio as input(s). The evaluation of the specified design variable(s) and/or response signal(s) within MATLAB produces the result(s) or output(s) engaged by ADE 1.0 for further evaluations. *(See the video tutorials for more details "ADE 1.0".)*

- *User-defined MATLAB Functions: DRA Test Case*

  - **Smart Parameters: Geometrical Constraint Function**

  To derive real parameters from smart parameters, subject to simple geometrical constraints, the automatically generated MATLAB function must be updated in ADE 1.0 as follows:

  In this instance, "ay0" has been constrained geometrically to be less than "0.5 × ay" for the DRA model. "x" is a vector holding all values for the seven DRA parameters. To obtain the real values of “ay0”, subject to the given constraint, the following mathematical evaluations must be fed into the ADE 1.0 optimization:

```
function x = deriveXforDRA(x)
```

The content of the 4th column, that is “ay0”, in the 1 by 7 vector holding the input parameters, is extracted and divided by 10 to obtain 10% of the values (its range was set to [0,5] and we require [0,0.5]):

```
ay0=x(4)/10;
```

The input variables are regenerated by creating a new vector, where the variables in the first columns, that is “ax”, “ay” and “az”, are unaltered; the variable in the 4th column, that is “ay0”, is derived by keeping its value within 50% of “ay”; and the last three columns, that is “us”, “ws” and “ys”, are also unaltered:

```
x= [x(1:3),x(2)*ay0,x(5:7)];
```

Note that when Smart Parameters are used, in Sample Verification, the true parameters (not smart parameters) have to be used in the 1.0 version.

- ***Response Templates and Interpolation***

  - For each of the defined and selected responses from the list of output signals from the CST simulation, a cell or matrix is generated with the operating frequencies (from the CST simulation) in the first column and the corresponding signal values (real and/or imaginary) in the respective columns. This can be observed by viewing the files of the response signals in the MATLAB workspace.

    ADE 1.0 ‘s response templates of are designed to allow the user to specify the operating frequency range required for the optimization process. To do this, the user must specify a ‘start frequency’ and a ‘stop frequency’ recommended to be within the bounds of the operating frequency range defined for the model in CST Microwave Studio. Using these ‘start frequency’ and ‘stop frequency’ values, the MATLAB functions in the response templates generate all possible response values within the specified frequency range. An interpolation can be carried out by the user, using the lower and upper frequency bounds specified, with a ‘frequency step-size’ also defined by the user. This can achieve a response at frequencies that do not exist in the simulation. A test case has been given as an example using the YUA problem in the video tutorial available through this link: CADES Research Center (2016), Yagi-Uda Antenna (YUA) Optimization Using ADE 1.0

    A detailed perspective of the interpolation process may be summarized as follows for an S-parameter response:

    The response and function have been defined as “s11” and “YUA_response” respectively.

```
function y = s11(x,YUA_response)
```

The array holding all the magnitude values and corresponding frequencies of the scattering parameter is ascribed to “S11_temp”. In this instance, the array is YUA_response {2} because the scattering parameter is the second response signal under consideration.

```
S11_temp=YUA_response{2};
```

The frequency range of the antenna is defined from 10GHz to 11GHz using a step size of 10MHz:

```
omega=10e9:10e6:11e9;
```

A frequency unit of 1GHz is specified to obtain exact frequency values within the frequency change:

```
Freq_Unit=1e9;
```

For the S-parameter in 'dB', the frequency values are in the 1st column and the real values are in the 2nd column. This can be observed by viewing the “.mat file” of the response signal in the workspace.

The real values of frequencies are extracted from the 1st column and assigned to “F”:

```
F=S11_temp(:,1);
```

The 'dB' values of the scattering parameter are extracted from the 2nd column and assigned to G:

```
S11=S11_temp(:,2);
```

A piecewise cubic interpolation (“pchip”) is carried out between “F” and “G”. The output of the interpolation is transposed to have a single row of values and assigned to “R”, which holds values corresponding to a very broad range of frequencies within the frequency range:

```
R=transpose(interp1(F/Freq_Unit,S11,omega/Freq_Unit,'pchip'));
```

Evaluation for optimization using the interpolated range of values between 10GHz and 11GHz is carried out:

```
y=max(R);
```

5.0 PROBLEM EXAMPLES

This section introduces some of the many complex antenna optimization problems, which ADE 1.0 has been used to address. The problem examples presented here are for the following antennas: Dielectric Resonator Antenna (DRA), Yagi-Uda Antenna (YUA), Broadband Microstrip Antenna (BMA), and Linear Microstrip Antenna Array (LMAA). It is strongly recommended to visit "ADE 1.0" tutorial videos for a full analysis and description of these examples.

5.1 Dielectric Resonator Antenna (DRA) Example

In this example, the design parameters of an aperture-fed DRA have been adjusted to meet the requirement of an operating frequency of 5.28GHz to 5.72GHz. The desired resonance frequency is 5.5GHz in the VSWR < 3 region. That is, the bandwidth is to be centered at 5.5GHz. The value of the fractional impedance bandwidth (FBW), that is the measure of the frequency band in which the DRA can be efficiently driven by a source at -10dB, should be no more than 8%. The backward radiation (losses) on the substrate should be as low as possible. With respect to radiation in the far-field of the DRA, the realized gain is not to be less than 3dB and -10dB for the zero zenith angle and back radiation respectively. Side/back lobes should be at least 25dB down from the main beam gain in the region 70° < Φ < 290°. The E-plane (the plane parallel to the plane of the antenna) and the H-plane (the plane perpendicular to the E-plane) should have beamwidths of about 50°.

Fig. 22 shows the geometry of a DRA. The design constraints imposed with respect to the radiation of the DRA are summarized as follows: the realized gain is to be not less than 3dB for the zero zenith angle and the realized gain of back radiation is to be less than -10dB. These design constraints are to be imposed over the impedance bandwidth achieved. Consequently, there is a single design objective function, which is to minimize the maximum reflection coefficient S_{11} from 5.28GHz to 5.72GHz:

$$\min \quad \max|S_{11}|$$

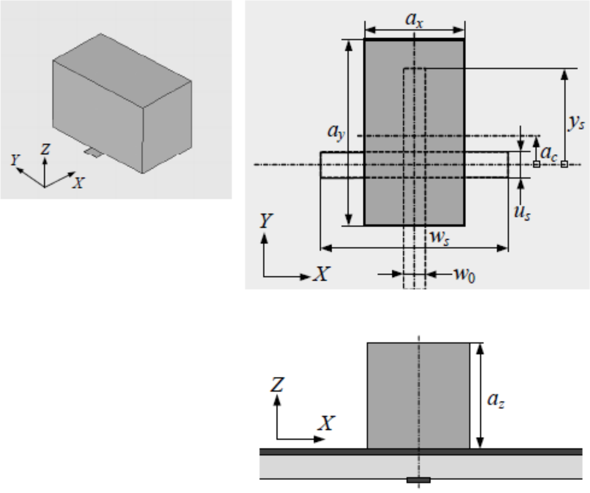


Fig. 22: Geometry of a Dielectric Resonator Antenna (DRA)

The range of design variables adopted for the optimization of the DRA can be found in Table 4:

Table 4: Range of Design Variables (all in mm) for Design Exploration of the DRA

Variables	ax	ay	ay0	ac	us	ws	ys
Lower Bound	6	12	6	6	0.5	4	2
Upper Bound	10	16	10	8	4	12	12

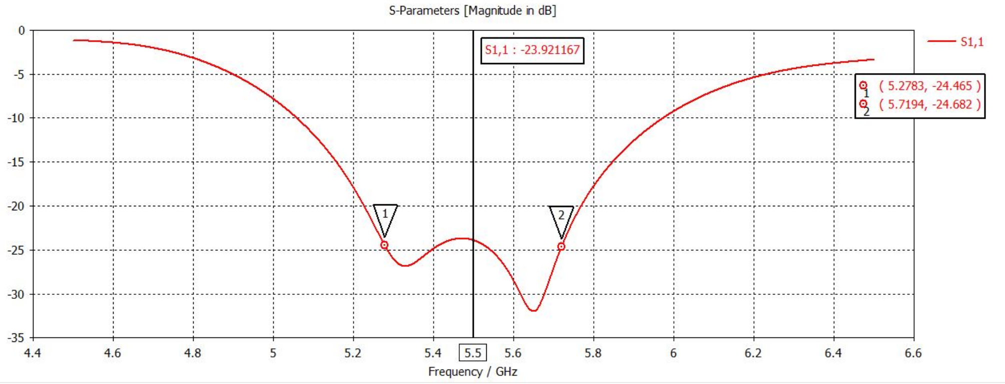


Fig. 23: Optimized S-Parameter Plot for the DRA

Fig. 23 shows CST simulation results after optimization of the DRA problem using ADE 1.0. The optimum reflection coefficient values at 5.28GHz, 5.5GHz and 5.72GHz are -24.27db, -23.92dB and -24.68dB respectively. These values validate realization of the optimization objective strongly. A full description, and step by step illustration, of the solution to the DRA optimization problem using ADE 1.0 has been made available by the CADES research center. This can be found using this link: CADES Research Center (2016), Dielectric Resonator Antenna (DRA) Optimization Using ADE 1.0

5.2 Yagi-Uda Antenna (YUA) Example

In this example, the design parameters of an 8-variable Yag-Uda antenna have been adjusted to meet the requirement of a bandwidth between 10GHz and 11GHz. The design objective is to minimize the maximum reflection coefficient, S_{11} as follows:

$$\min \quad \max|S_{11}|$$

Fig. 24 shows the geometry of a YUA. The design objective is subject to a constraint, which stipulates that the magnitude of the average gain should not be smaller than 6 in the frequency region of 10GHz to 11GHz. This constraint is expressed mathematically as follows:

$$\min \quad \max|S_{11}|$$

$$s.t. \quad mean(G) \geq 6$$

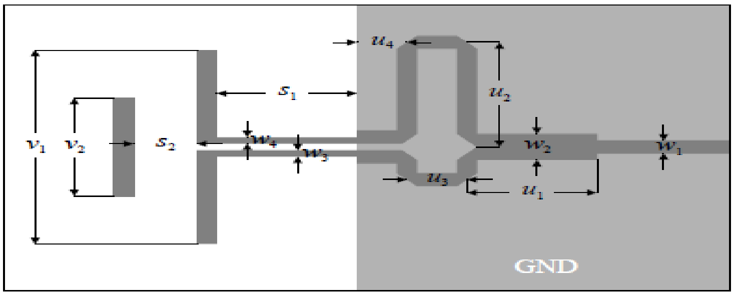


Fig. 24: Geometry of a Yagi-Uda Antenna (YUA)

The range of design variables adopted for the optimization of the YUA can be found in Table 5:

Table 5: Range of Design Variables (all in mm) for the Design Exploration of the YUA

Variables	s_ref	s_dir	u_d	u_dir	u1	u3	u4	u5
Lower Bound	3	1	5	2	2	2	1	1
Upper Bound	7	6	12	12	6	6	5	5

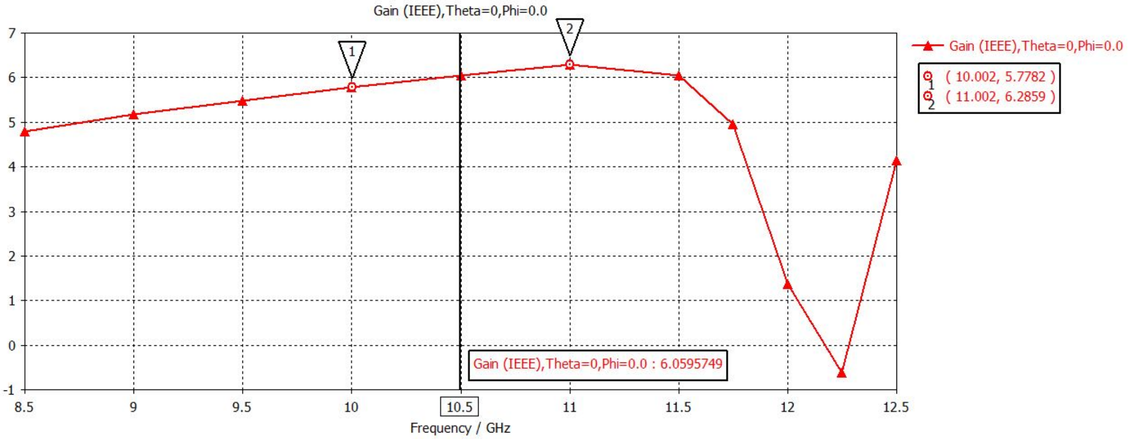


Fig. 25: Optimized Gain Plot for the YUA

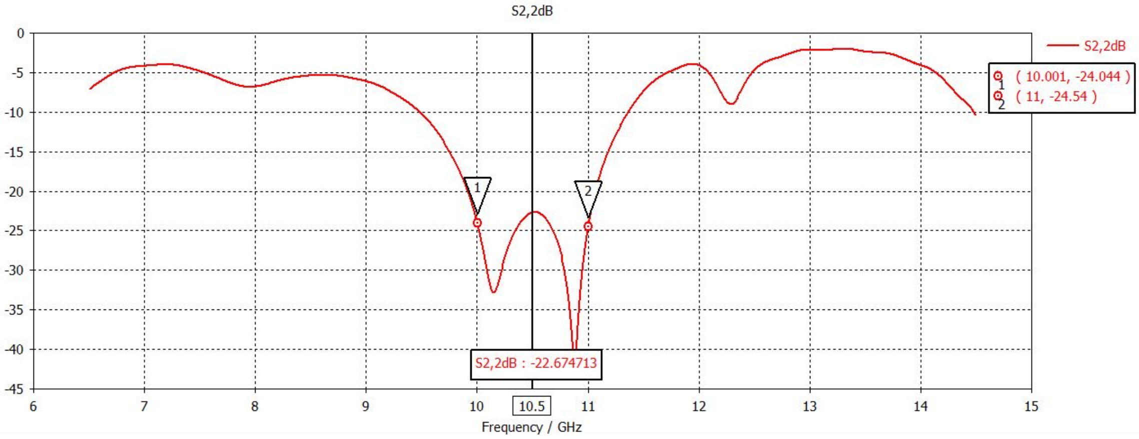


Fig. 26: Optimized S-Parameter Plot for the YUA

Fig. 25 and Fig. 26 show CST simulation results after optimization of the YUA problem using ADE 1.0. The optimum gain magnitudes at 10GHz, 10.5GHz and 11GHz are 5.78, 6.06 and -6.29 respectively, while the optimum reflection coefficient values at 10GHz, 10.5GHz and 11GHz are -24.04dB, -22.67dB and -24.54dB respectively. These values validate realization of the optimization objective strongly. A full description, and step by step illustration, of the solution to this optimization problem using ADE 1.0 has been made available by the CADES research center. This can be found using this link: CADES Research Center (2016), Yagi-Uda Antenna (YUA) Optimization Using ADE 1.0

5.3 Broadband Microstrip Antenna (BMA)

In this example, the design parameters of the BMA have been adjusted to meet the requirement of a bandwidth between 2.6GHz and 5.25GHz. The design objective is to minimize the maximum reflection coefficient, S_{11} from 3.1GHz to 4.8GHz as follows:

$$\min \quad \max|S_{11}|$$

Fig. 27 shows the 3-D geometry of a BMA. Owing to a large range of variables, the design objective for the BMA problem is subject to the following geometrical constraints:

$$x_3 + 2 - x_5$$

$$x_5 - 2 \times \min\left(\left[30 - {x_3}/{2} - 5 \quad {x_4}/{2} - 5\right]\right)$$

$$-{x_1}/{2} + abs(x_6)$$

$$x_8 - \min\left(\left[{x_1}/{2} \quad {x_2}/{2} \quad {x_3}/{2} \quad {x_4}/{2}\right]\right)$$

where $x_i \leq 0 \quad i = 1:1:8$

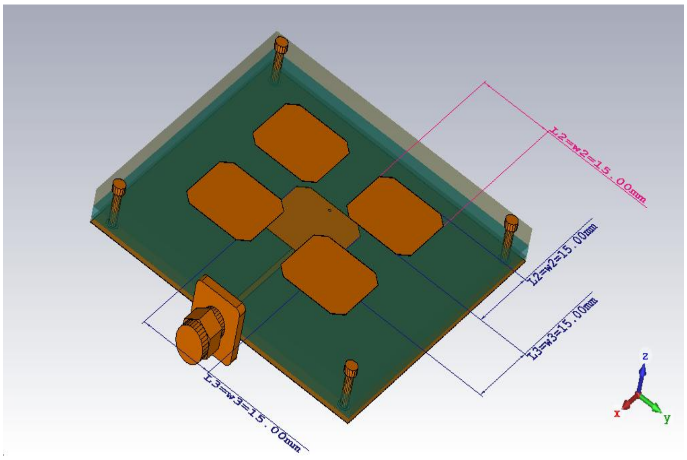


Fig. 27: Geometry of a Broadband Micro-strip Antenna (BMA)

The range of design variables adopted for the optimization of the BMA can be found in Table 6

Table 6: Range of Design Variables (all in mm) for the Design Exploration of the BMA

Variables	L2	w2	L3	w3	d3	x_via	dx_stub	q2
Lower Bound	12	12	10	10	12	-9	0.25	0.5
Upper Bound	18	18	20	20	40	9	30	10
Notation: Constraint Equations	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8

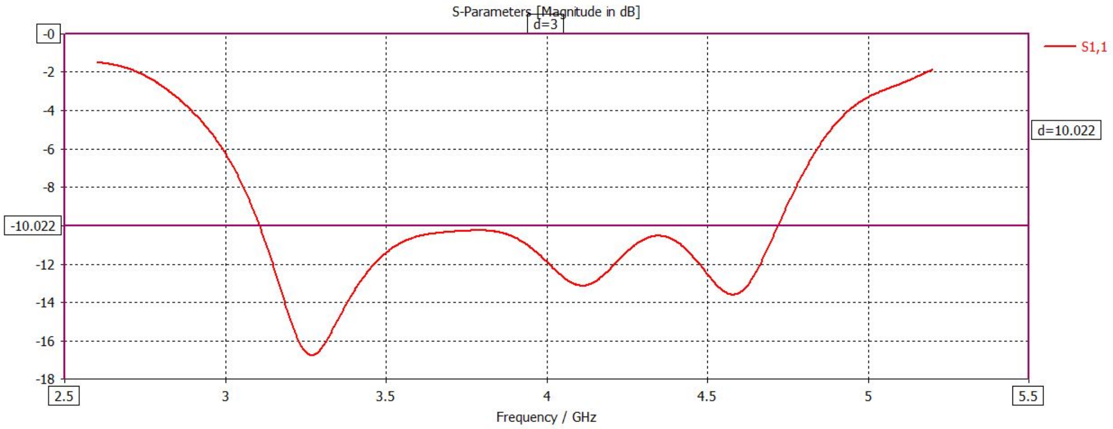


Fig. 28: Optimized S-Parameter Plot for the BMA

Fig. 28 shows CST simulation results after optimization of the BMA problem using ADE 1.0. Using the -10dB reference, it can be seen that the BMA clearly exhibits broadband characteristics over the desired bandwidth. This validates the realization of the optimization objective strongly. A full description, and step by step illustration, of the solution to the BMA optimization problem using ADE 1.0 has been made available by the CADES research center. This can be found using this link: CADES Research Center (2016), Broadband Microstrip Antenna (BMA) Using ADE 1.0

5.4 Linear Microstrip Antenna Array (LMAA)

In this example, the design parameters of a 16-variable LMAA have been adjusted to meet the requirement of an operating frequency at 10GHz. The design objective is to minimize the level of the side lobes assuming ±8^{0} of the main beam, that is, S_{LL} is expressed as follows:

$$\min \quad |S_{LL}|$$

Fig. 29 shows the geometry of a Linear Microstrip Antenna Array (LMAA). With respect to the design objective, S_{LL} will be the maximum relative power for the angles 0^{0} to 82^{0} and 98^{0} to 180^{0}.

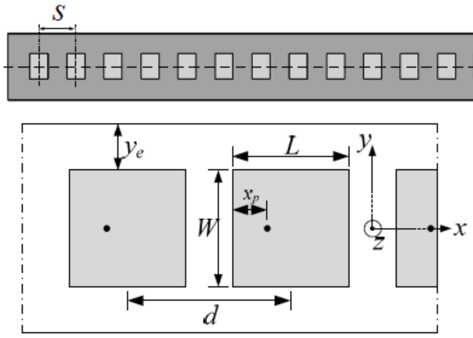


Fig. 29: Geometry of a Linear Micro-strip Antenna Array (LMAA)

For the LMAA problem, there 16 design variables, which are excitation amplitudes a_k: k = 1:1:16 with a range of $[0; 1]^{16}$.

A full description, and step by step illustration, of the solution to the LMAA optimization problem using ADE 1.0 has been made available by the CADES research center. This can be found using this link: CADES Research Center (2016), Linear Microstrip Antenna Array (LMAA) Using Analytical Formula on ADE 1.0

REFERENCES

[1]. B. Liu, H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen and P. Excell. An Efficient Method for Antenna Design Optimization Based on Evolutionary Computation and Machine Learning Techniques (2014), *IEEE Transactions on Antennas and Propagation*. vol. 62(1), pp. 7 – 18.

[2]. Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition (2007), *IEEE Transactions on Evolutionary Computation*. vol. 11(6), pp. 712 – 731.

[3]. R. Carvalho, R. R. Saldanha, B. N. Gomes, A. C. Lisboa and A. X. Martins. A Multi-Objective Evolutionary Algorithm Based on Decomposition for Optimal Design of Yagi-Uda Antennas (2012), *IEEE Transactions on Magnetics*. vol. 48(2), pp. 803 – 804.

[4]. M. O. Akinsolu, A. H. Ali, A. Atojoko, E. Ibrahim, I. T. Elfergani, R. A. Abd-Alhameed, A. S. Hussaini, J. M. Noras and J. Rodriguez (2015), "Novel Quadrifilar Helical Antenna in RFID Applications Using Genetic Algorithms", *Progress in Electromagnetics Research Symposium (PIERS) Proceedings*, pp. 528 – 532.

[5]. B. Liu, G. G. E. Gielen and F. V. Fernandez (2014), "Basic Concepts and Background". In Automated Design of Analog and High-frequency Circuits: A Computational Intelligence Approach. Berlin: Springer-Verlag Berlin Heidelberg, pp. 1 – 16.

[6]. A. Hedar, (2005, Nov.), *Test Problems for Constrained Global Optimization: G12 Problem.* [Online] Available at:http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page4221.htm

[7]. A. Ibrahim, S. Rahnamayan, M. V. Martin and K. Deb (2016), "3D-RadVis: Visualization of Pareto Front in Many-Objective Optimization", *IEEE World Congress on Computational Intelligence (IEEE WCCI) Proceedings*, pp. 736-745.

[8]. C. Noilublao and S. Bureerat (2009), "Simultaneous Topology, Shape and Sizing Optimisation of Skeletal Structures Using Multiobjective Evolutionary Algorithms". In Evolutionary Computation. Vienna: I – Tech, pp. 487 – 508.

[9]. A. E. Smith and D. W. Coit (1996), "Penalty Functions". In Handbook of Evolutionary Computation. T. Baeck, D. Fogel and Z. Michalewicz Ed. Bristol: Institute of Physics Publishing and Oxford University Press, section C 5.2.

[10]. B. Liu, S. Koziel and N. Ali (2016), SADEA-II: A Generalized Method for Efficient Global Optimization of Antenna Design, *Journal of Computational Design and Engineering*, Elsevier, (In press). Available: http://www.sciencedirect.com/science/article/pii/S2288430016300859

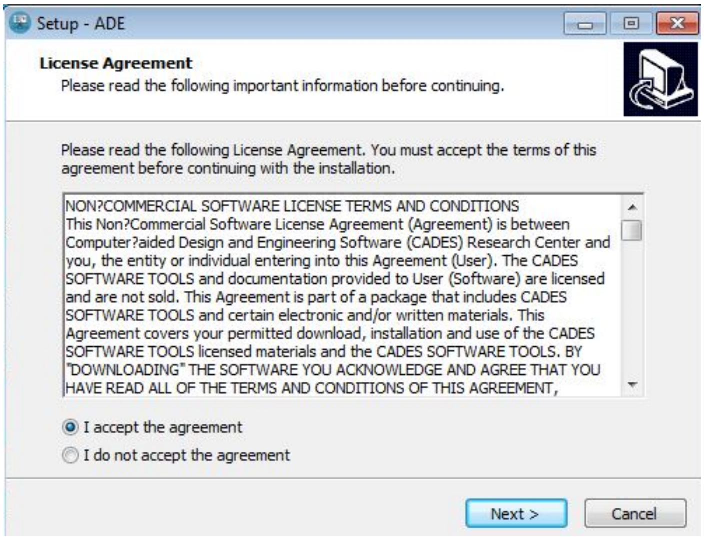
[11]. R. Storn and K. Price (1997), Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, Springer, vol. 11, no. 4, pp. 341 – 3599.

APPENDIX 1: ADE INSTALLATION QUICK GUIDE

- **To install the software, you need to download and run the setup file on your PC. The icon for the setup file is shown in Fig. 30:**

Fig. 30: ADE Setup Icon

- **As shown in Fig. 31, the 'End User License Agreement' must be accepted to continue the installation process:**



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

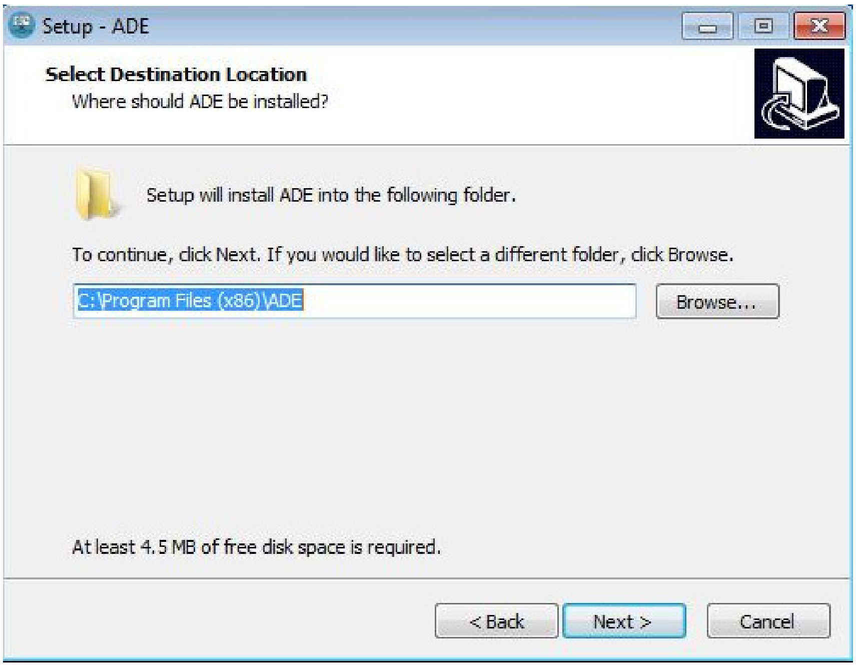
NON?COMMERCIAL SOFTWARE LICENSE TERMS AND CONDITIONS
This Non?Commercial Software License Agreement (Agreement) is between Computer?aided Design and Engineering Software (CADES) Research Center and you, the entity or individual entering into this Agreement (User). The CADES SOFTWARE TOOLS and documentation provided to User (Software) are licensed and are not sold. This Agreement is part of a package that includes CADES SOFTWARE TOOLS and certain electronic and/or written materials. This Agreement covers your permitted download, installation and use of the CADES SOFTWARE TOOLS licensed materials and the CADES SOFTWARE TOOLS. BY "DOWNLOADING" THE SOFTWARE YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE READ ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT,

● I accept the agreement

○ I do not accept the agreement

Fig. 31: ADE End User License Agreement (EULA)

- By default, the ADE is installed on the path of executable files on the PC. However, a different or unique location or path can be specified on the PC by using the "Browse" button. Fig. 32 can be referred to.



Select Destination Location

Where should ADE be installed?

Setup will install ADE into the following folder.

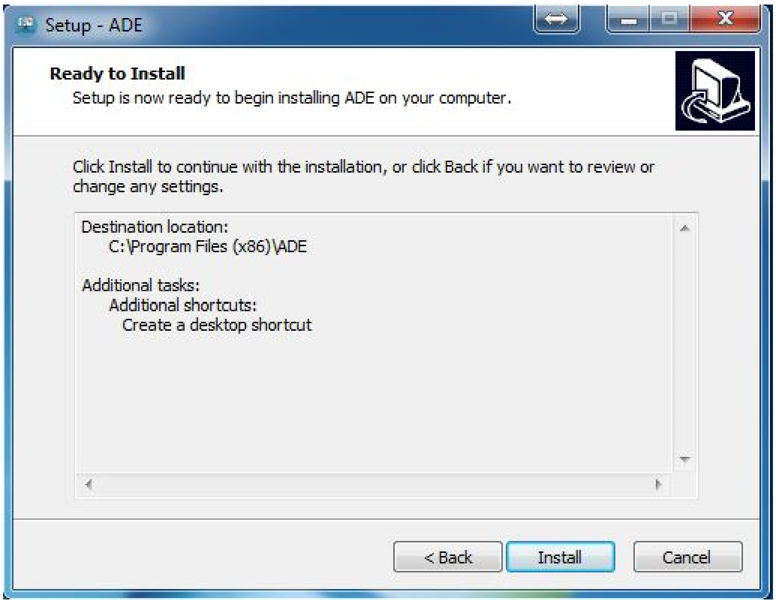
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\ADE

At least 4.5 MB of free disk space is required.

Fig. 32: ADE Installation Location / Path

- Once the installation location or path has been specified, the installation settings are confirmed by clicking the "Install" button. To change or edit installation settings, the "Back" button can be clicked. Fig. 33 can be referred to.



Ready to Install

Setup is now ready to begin installing ADE on your computer.

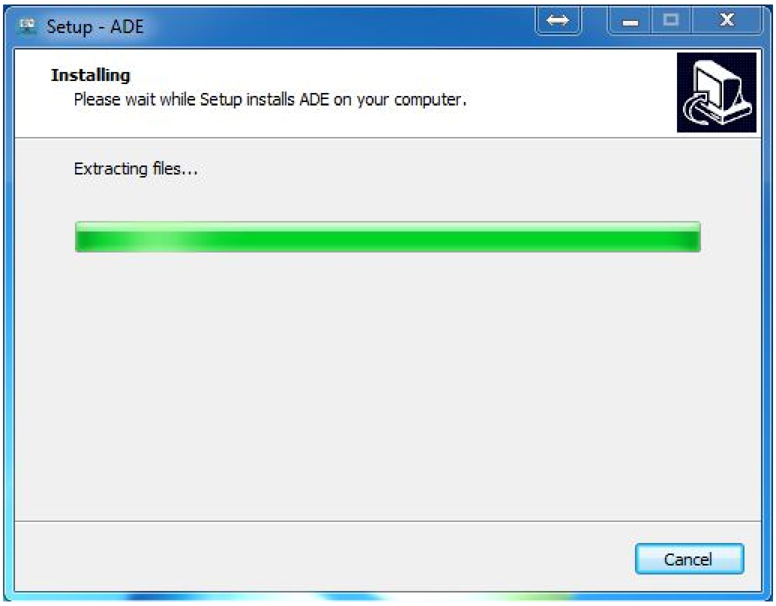
Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files (x86)\ADE

Additional tasks:
    Additional shortcuts:
      Create a desktop shortcut
```

Fig. 33: ADE Installation Settings Confirmation

- **After clicking the “Install” button, the ADE installation commences automatically as shown in Fig. 34.**



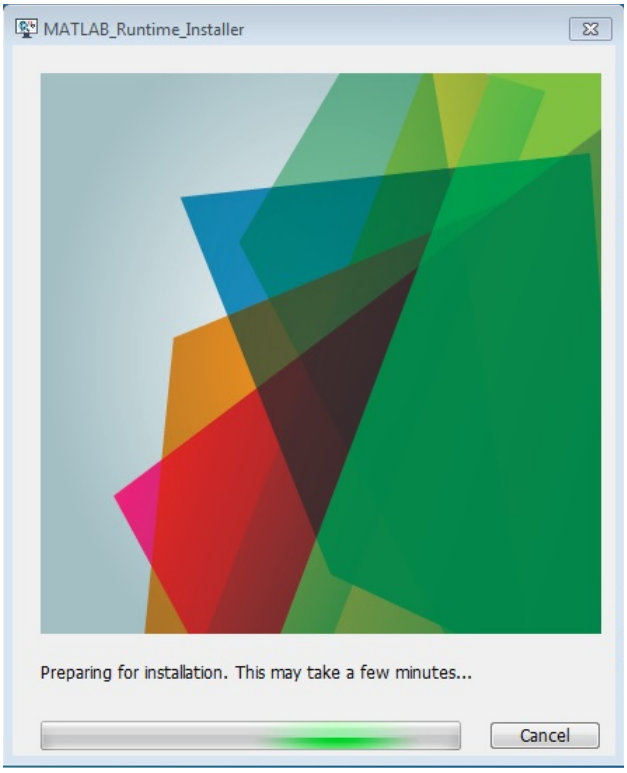
Installing

Please wait while Setup installs ADE on your computer.

Extracting files...

Fig. 34: ADE Installation Commencement

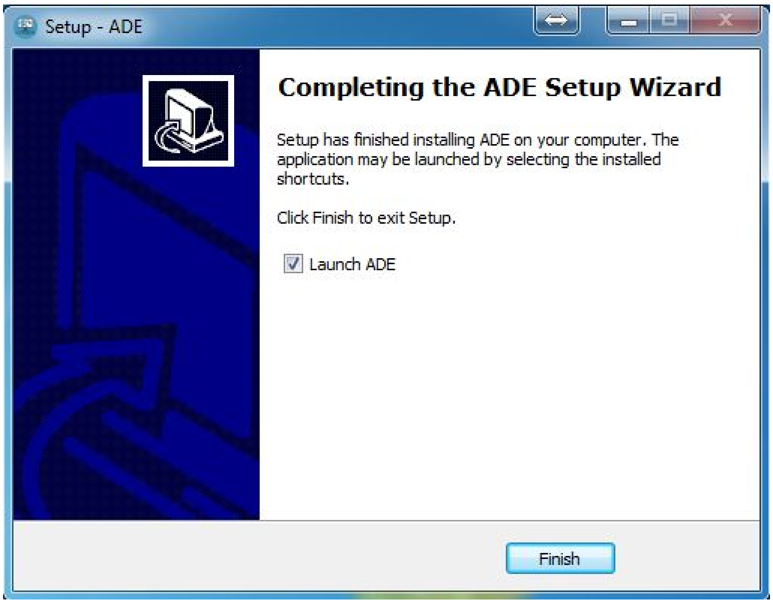
- **The run of the ADE’s setup file is packed to also install a compatible version of the “MATLAB Runtime” automatically. Fig. 35 can be referred to.**



Preparing for installation. This may take a few minutes...

Fig. 35: ADE MATLAB Runtime Installation

- **Once a compatible version of the "MATLAB Runtime" has been installed, the installation process of the ADE is completed by clicking the "Finish" button as shown in Fig. 36.**



Completing the ADE Setup Wizard

Setup has finished installing ADE on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch ADE

Fig. 36: ADE Installation Completion