

OptiGein Design

User Manual

Design of Experiments software

Version 1.5.0

Geinforce Technology Private Limited

info@geinforce.com | +91 9960867592 | geinforce.com

OptiGein Design User Manual

Software version 1.5.0

Published by Geinforce Technology Private Limited. This manual describes OptiGein Design, a Design of Experiments application for Windows and Linux. It is generated from the product documentation, so it always matches the software release named above.

Support: info@geinforce.com
Telephone: +91 9960867592
Web: geinforce.com

Copyright 2026 Geinforce Technology Private Limited. All rights reserved. No part of this manual may be reproduced or distributed without written permission. Product names mentioned belong to their respective owners.

Contents

Getting started **9**

Introduction . . . 9

Start here . . . 9

What the software does . . . 9

How this guide is organised . . . 12

Getting help . . . 12

Installing the software . . . 14

Before you start . . . 14

Installing on Windows . . . 14

Installing on Linux . . . 15

macOS . . . 16

Keeping the software up to date . . . 16

Next: activate your copy . . . 18

License and activation . . . 19

What you need before you start . . . 19

Activating on first launch . . . 19

What activation does to the device . . . 21

Working offline . . . 21

When a license expires . . . 22

Checking your license status . . . 23

Moving to a new computer . . . 23

Messages you might see . . . 23

Getting help . . . 24

A tour of the software . . . 25

The welcome screen . . . 25

Choosing a design . . . 26

The run sheet . . . 28

The navigation pane . . . 29

Analysing a response . . . 30

Seeing the model . . . 30

Finding the optimum . . . 31

What to read next . . . 32

Choosing a design 33

Which design do I need? . . . 33

Three stages, not sixteen decisions . . . 33

Mixture designs are a special case . . . 34

The full catalogue . . . 34

Let the wizard choose for you . . . 36

Go deeper on a family . . . 37

Factorial designs . . . 38

Resolution, and why the two-level grid is coloured . . . 38

Regular Two-Level . . . 38

Multilevel Categoric . . . 39

Plackett-Burman . . . 40

Min-Run Characterize (Resolution V) . . . 41

Min-Run Screen (Resolution IV) . . . 42

Taguchi OA . . . 43

Center points, replication and run order . . . 44

Response surface designs . . . 45

Why you need more than two levels . . . 45

Central composite . . . 45

Box-Behnken . . . 47

Optimal (Custom) . . . 48

Definitive screening . . . 48

What to read next . . . 49

Mixture designs . . . 50

What makes a mixture different . . . 50

Why a factorial does not work here . . . 50

How a mixture model is fitted . . . 50

Setting a minimum and a maximum for a component . . . 51

Simplex Lattice . . . 52

Simplex Centroid . . . 53

Optimal (Custom) . . . 53

Reading the run sheet . . . 54

What to read next . . . 55

Custom designs and import . . . 56

Optimal (Combined) . . . 56

Blank Spreadsheet . . . 57

Import Data Set . . . 57

Working with a design **60**

The design workspace . . . 60

The run sheet . . . 60

Editing factors and responses . . . 61

The Information branch . . . 62

Saving your work . . . 65

Evaluating a design . . . 67

Choosing the model to evaluate . . . 67

Degrees of freedom . . . 67

Leverage and average leverage . . . 68

Per-term precision: standard error, VIF and Ri squared . . . 68

Power: the chance of detecting an effect . . . 69

Design quality measures and prediction variance . . . 69

The fraction of design space curve . . . 70

When the numbers look bad . . . 70

Analysing results **71**

Fit summary, model and ANOVA . . . 71

Configure: set up the fit . . . 71

Fit Summary: which model does the data support? . . . 72

Model: choose the exact terms . . . 73

ANOVA: read the verdict . . . 75

The two equations, and when to use each . . . 76

Diagnostics . . . 79

Normal plot of residuals . . . 80

Residuals versus predicted, run and factor . . . 80

Box-Cox plot for a transform . . . 80

Predicted versus actual . . . 81

Influence: who is steering the model . . . 81

What to do when a point is flagged . . . 82

The Report table . . . 82

Model graphs . . . 84

The Factors Tool: axes and slices . . . 84

Contour . . . 85

3D Surface . . . 86
Interaction . . . 86
Perturbation . . . 87
One Factor . . . 88
Pareto . . . 89
Predicted versus Actual . . . 90
Copying and exporting any graph . . . 91

Finding the best settings **92**

Numerical optimization . . . 92
Setting a goal for each factor and response . . . 92
Weights and the shape of the ramp . . . 94
Importance . . . 94
How the scores combine . . . 94
The Solutions table . . . 95
The Graphs tab . . . 95
Picking a solution and confirming it . . . 97
Graphical optimization . . . 98
Why look at a picture . . . 98
Set the limits . . . 98
Read the overlay . . . 99
When you have more than two factors . . . 100
When there is no yellow region . . . 101
Prediction and confirmation . . . 102
Point Prediction . . . 102
Confirmation . . . 104
Coefficients Table . . . 106
Where this fits . . . 107

Tutorials **108**

Response surface, start to finish . . . 108
What you learned . . . 118
Where to go deeper . . . 119
Screening with a factorial . . . 120
Why screen first . . . 120
The example . . . 120

Step by step . . . 120
Reading the result . . . 124
What to do next . . . 124
A mixture experiment . . . 125
When the ingredients are the experiment . . . 125
Build the design . . . 125
The blends the software builds . . . 126
Enter your results . . . 127
Fit the mixture model . . . 127
Read the surface . . . 129
Predict any blend, and the pure components . . . 130
What to read next . . . 131

Reference **132**

Saving, printing and exporting . . . 132
Saving your work as a design file . . . 132
Opening a design again . . . 132
Exporting the run sheet to a spreadsheet . . . 133
Printing and print preview . . . 133
Copying and exporting a graph . . . 133
Copying and exporting a results table . . . 134
What to read next . . . 135
Menus and shortcuts . . . 136
File menu . . . 136
Edit menu . . . 137
View menu . . . 138
Display Options menu . . . 138
Design Tools menu . . . 138
Help menu . . . 139
Toolbar buttons . . . 139
Glossary . . . 141
Troubleshooting . . . 147
The app will not start . . . 147
Updates and appearance . . . 148
Analysis . . . 149
Optimization . . . 150

Graphs and export 151
Contacting support 152

Part 1

Getting started

Chapter 1

Introduction

Plan an experiment, run it, and let the data tell you where the best settings are. This guide walks through every screen of the software, from picking a design to reporting the optimum you found.

OptiGein Design is a Design of Experiments (DOE) application for Windows and Linux, built by Geinforce Technology Private Limited. It covers the whole cycle of an experiment: choosing a design that answers your question with as few runs as possible, entering the results, fitting and checking a model, and then reading the optimum settings off that model.

New to Design of Experiments?
Start with A tour of the software, then follow the response surface tutorial. Between them they use every screen you are likely to need, on one small worked example.

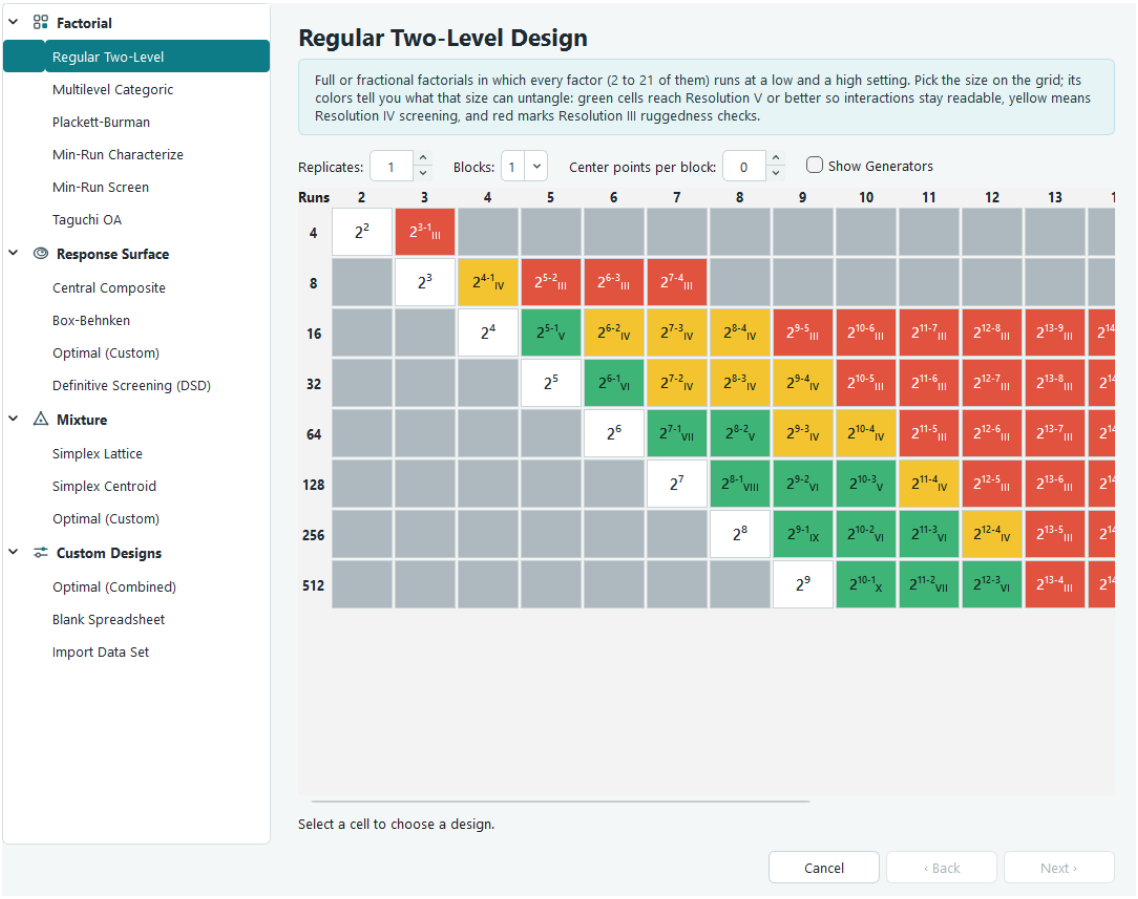
Prefer to read offline or print it?
The whole guide is also available as a single PDF user manual, with a contents list, an index and every screenshot. It is generated from these pages, so the two never disagree.

Start here

What the software does

Build a design

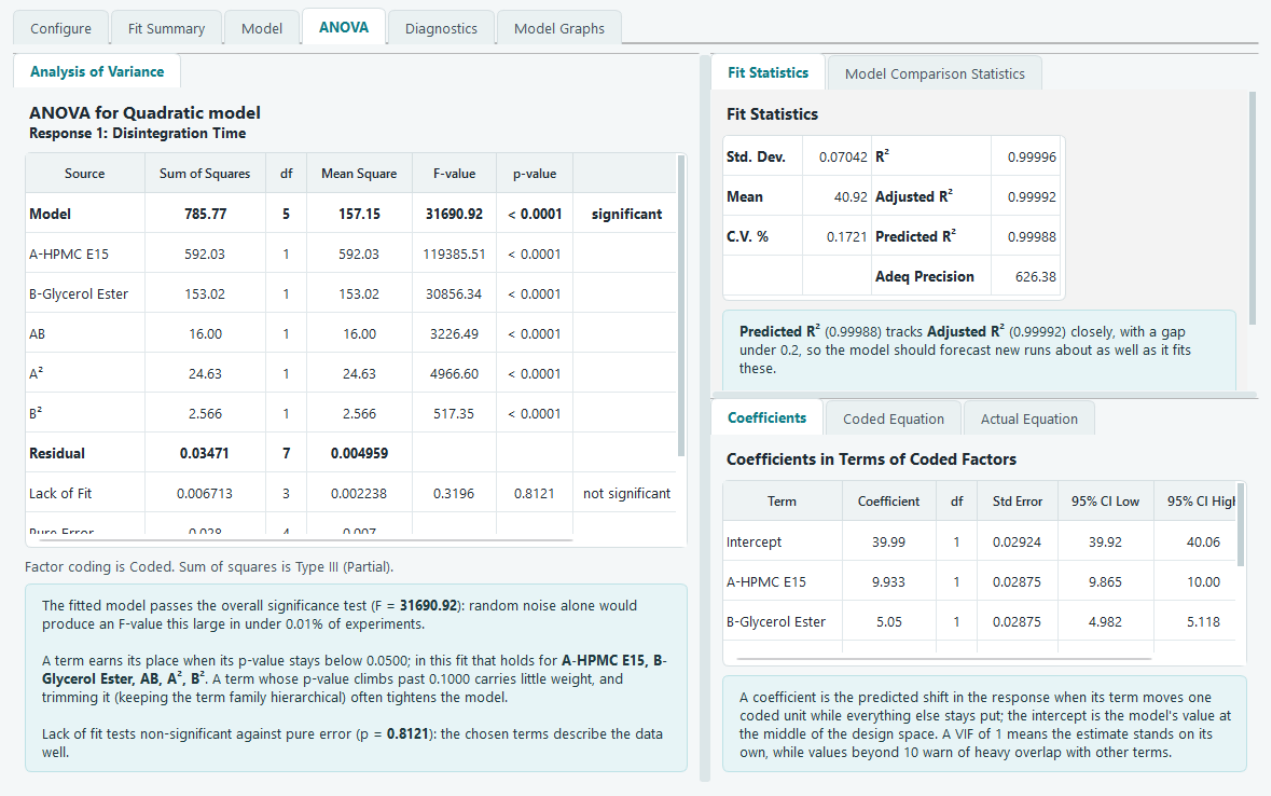
Sixteen design types are grouped into four families. Factorial designs sort the important factors from the unimportant ones. Response surface designs map curvature and locate an optimum. Mixture designs handle recipes, where the ingredients must add up to a fixed total. Custom designs cover everything else, including a blank sheet and importing data you already have.



Every design the software can build, grouped by what it is good for.

Analyse the results

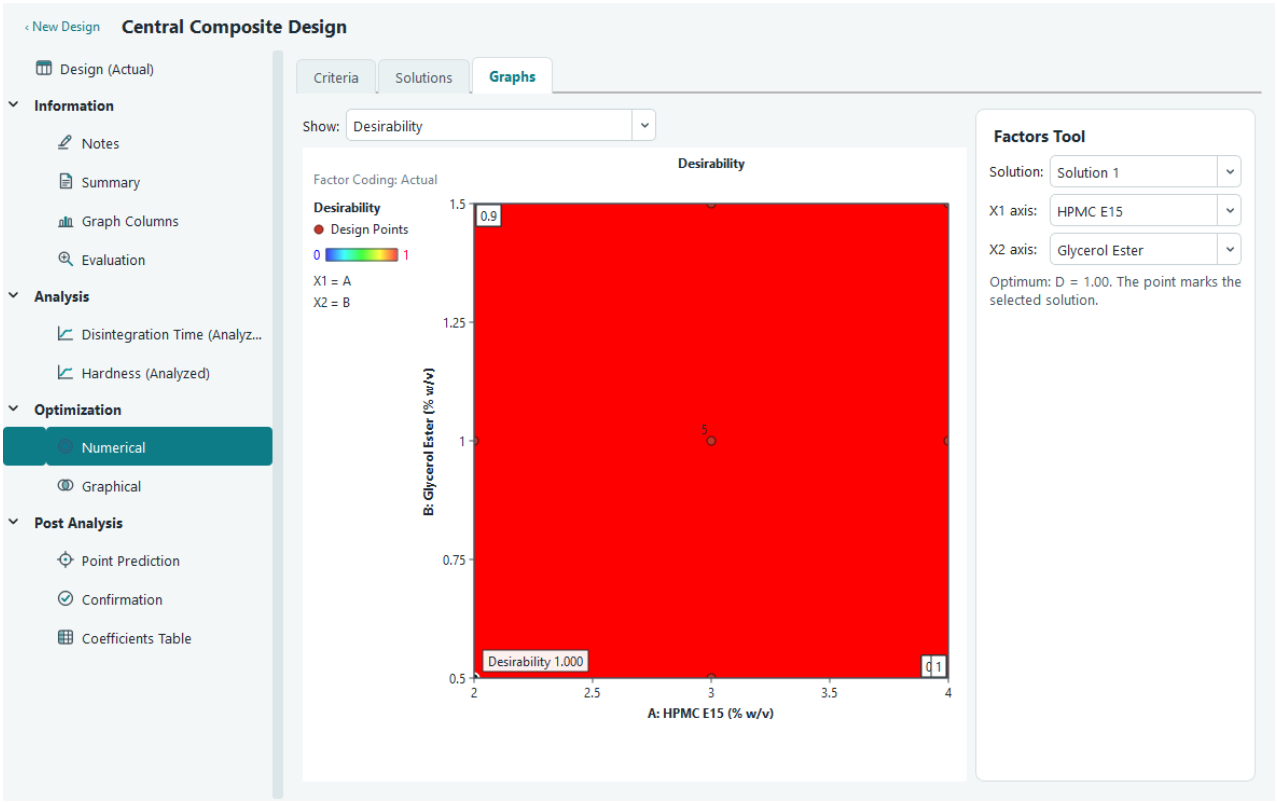
Type your measurements into the run sheet, press Start Analysis, and the software fits a model, suggests the order that the data supports, and lays out the ANOVA table with plain language notes telling you what each number means. Residual diagnostics tell you whether the model can be trusted.



The ANOVA table, fit statistics and the fitted equation, all on one screen.

Find the best settings

Set a goal for each response, such as maximise yield while holding cost in range, and the optimizer searches the fitted models for the settings that satisfy all of them at once. The graphical route shades the region where every response stays inside its limits.



The optimum marked on the response contours.

How this guide is organised

If you want to	Read
Install and activate the software	Installing and Activation
Understand what each screen is for	A tour of the software
Decide which design fits your question	Which design do I need?
Look up the settings of one design type	The four design pages, starting with Factorial
Read an ANOVA table with confidence	Fit summary, model and ANOVA
Check whether your model is trustworthy	Diagnostics
Explain the model to somebody else	Model graphs and Exporting
Get numbers for the best settings	Numerical optimization
Follow a complete worked example	Response surface tutorial
Watch it done on screen	Video tutorials
Move to a newer version	Keeping the software up to date

Getting help

If something in the software does not behave the way this guide describes, please tell us. Include the design type and, if you can, the saved `.ogdx` file.

- Email: info@geinforce.com
- Phone: +91 9960867592

- Web: geinforce.com

Chapter 2

Installing the software

OptiGein Design comes as one download per platform, and each download already contains everything the program needs to run. This page walks through the install on Windows and Linux, and shows how to remove it again.

Before you start

You do not need to install anything else first. The download carries its own copy of the Qt runtime the program is built on, and on Windows it also carries the Microsoft C++ runtime. So there is no separate Qt install, no Visual C++ redistributable and no other dependency to chase down. On Linux, the few system libraries the program uses, for the display, fonts, printing and secure connections, come from your distribution, and the package manager installs any that are missing. The application is 64-bit only.

Platform	You need	Download
Windows	64-bit Windows 10 or later	`OptiGein-Design-<version>-Windows-x64-Setup.exe`
Linux	64-bit Ubuntu 22.04 or Debian 12, or newer	`optigein-design_<version>_amd64.deb`
macOS	Not available yet	

Installing on Windows

The Windows download is a single self-extracting setup file. When you run it, a small stub unpacks the compressed payload, the program, its Qt runtime and the setup wizard, into a temporary folder, then starts the wizard from there. Nothing is written to your system until you reach the Install step, so you can cancel at any point before that and leave the machine untouched.

> **Administrator approval.**
> Setup installs the program for everyone on the machine, into Program Files, and records it in the machine-wide part of the registry. Because of that, Windows shows a User Account Control prompt when setup starts. Approve it to continue. On a managed work computer without administrator rights, ask whoever looks after the machine to run the installer for you.

The wizard has five short pages. Here is what each one asks of you.

1. **Start the installer.** Double-click the setup file and approve the Windows administrator prompt. The wizard opens with a title banner that shows the version you are about to install.
2. **Welcome.** The first page tells you what you are installing. There is nothing to fill in, so read it and click *Next* .
3. **License Agreement.** The full license text is shown in a scrolling box. Tick *I accept the terms of the license* to turn on the *Next* button, then click *Next* . If you do not accept, you cannot go on.
4. **Install Location and Options.** This is the only page with real choices. The install folder defaults to `C:\Program Files\OptiGein Design` ; use *Browse* to pick another folder if you prefer. If the folder you pick already holds other files, setup installs into an `OptiGein Design` folder inside it, so your files are never mixed with the program's. Below it are four options, all on by default: The page also shows the disk space the install needs. When the options look right, click *Install* .

Option	What it does
Create a Start Menu shortcut	Adds OptiGein Design to the all-users Start Menu.
Create a Desktop shortcut	Puts a shortcut on the shared desktop for every account.
Open .ogdx design files with OptiGein Design by default	Makes the program the default for the `.ogdx` design format, so a double-click opens the design.
Launch OptiGein Design when setup finishes	Opens the program as soon as you click Finish.

5. **Installing.** A progress bar tracks the work as setup copies the files, creates the shortcuts you asked for and registers the `.ogdx` file type. This takes a few seconds. You cannot cancel once copying has begun.
6. **Setup Complete.** The last page confirms where the program was installed and reminds you that you can start it from the Start Menu or the desktop shortcut, and remove it later from Windows. Click *Finish* . If you left the launch option ticked, the program opens right away.

Where the program is installed

By default the program lands in `C:\Program Files\OptiGein Design`. That folder holds the main program, `OptiGeinDesign.exe`, a small helper that doubles as the uninstaller, the Qt runtime libraries, the C++ runtime files and a copy of the license. The shortcuts, if you kept them, go to the all-users Start Menu and the shared (Public) desktop, so every account on the machine sees them. Setup also adds an entry to the Windows list of installed apps so you can remove the program cleanly later.

Opening designs by double-click

OptiGein Design saves your work in files with the `.ogdx` extension. If you left the file-association option ticked during setup, Windows now knows that `.ogdx` files belong to OptiGein Design. From then on, double-clicking any design file opens it straight in the program, and those files show the OptiGein Design icon in File Explorer.

Uninstalling on Windows

Open *Settings*, go to *Apps*, then *Installed apps* (on older Windows this is *Programs and Features* in Control Panel). Find OptiGein Design in the list and choose *Uninstall*. Windows asks for administrator approval again, the uninstaller asks you to confirm once, and then it removes the shortcuts, the `.ogdx` association, the registry entry and the program files. From version 1.5.0, the default folder in Program Files is removed whole; anywhere else, only the files setup put there are removed, and anything else in that folder is left alone.

> **Close the program first.**
> If OptiGein Design is running while you try to install over it or uninstall it, Windows cannot replace or delete the files in use. Close the program, then run setup or the uninstaller again.

Installing on Linux

From version 1.5.0, OptiGein Design runs on Linux. It comes as a Debian package (a `.deb` file) for 64-bit Ubuntu 22.04 or Debian 12 and newer, and for distributions built on them. The package installs the program with its own Qt runtime under `/opt/optigein-design`, adds it to the applications menu and registers the `.ogdx` design format.

1. **Install the package.** Open a terminal in the folder you downloaded it to and run `sudo apt install ./optigein-design_<version>_amd64.deb`, with the version of your download in place of `<version>`. Keep the `./` in front of the file name: it tells apt to install that file instead of searching for the name online. apt asks for your password and installs any system library the program needs that is missing.
2. **Start it.** Choose OptiGein Design in the applications menu, or run `optigein-design` in a terminal. Double-clicking an `.ogdx` file in the file manager opens it in the program.

> **Wayland desktops.**
> OptiGein Design draws its windows through X11. On a Wayland desktop, the standard on recent Ubuntu and Debian, it runs through XWayland, which those desktops already include, so there is nothing to set up.

Uninstalling on Linux

Run `sudo apt remove optigein-design`. This removes the program, its menu entry and the `.ogdx` file type. Your license and settings, which are kept in your home folder, are not removed, so installing the package again later does not need a new activation.

macOS

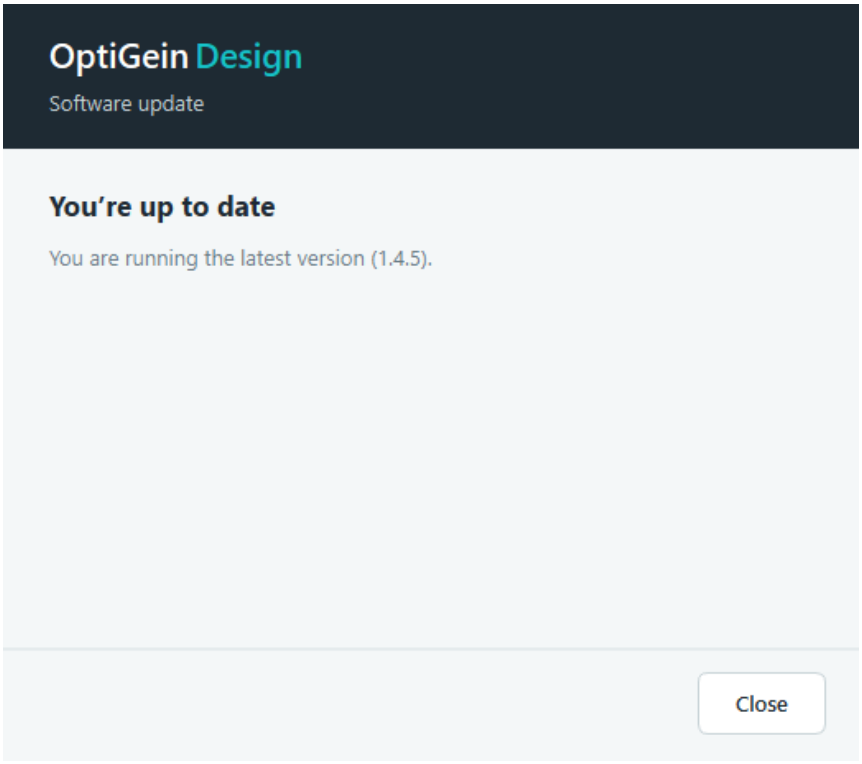
A macOS version of OptiGein Design is not available yet. If you need it, contact Geinforce and we will let you know when it is ready.

Keeping the software up to date

OptiGein Design does not update itself behind your back. You decide when to look for a new version, and nothing is downloaded until you say so. An update keeps your license, your settings and your saved designs.

Updating on Windows

1. Save your work, then choose **Help › Check for Updates**. The software asks the Geinforce server for the latest version. This needs an internet connection.



The Software Update window when there is nothing newer to install.

2. If you already have the latest version, the window says *You're up to date* and you can close it. If a newer one exists, it says *Update available* and shows the new version number, your current one, the download size and a list of what changed.
3. Click **Update Now** . A progress bar follows the download. When it finishes, the software checks the file against the fingerprint published by Geinforce, so a damaged or altered download is thrown away instead of being run. Click *Later* if you would rather not update yet; you can check again whenever you like.
4. When the window says *Ready to install* , click **Install & Restart** . OptiGein Design closes, Windows asks for administrator approval once, the new version is installed into the default Program Files folder with no wizard to click through, and the program opens again.

> **A required update.**
> Occasionally an update is marked as required. The window then says so, and warns that some features may not work until you install it.

Updating on Linux

On Linux the package manager looks after the installation, so the program downloads the update for you but you install it with apt.

1. Choose **Help › Check for Updates** . If a newer version exists, the window says *Update available* , as on Windows.
2. Click **Download Update** . The package is downloaded, checked against the fingerprint published by Geinforce, and saved in your *Downloads* folder.
3. The window says *Update downloaded* and shows the command that installs it, starting with `sudo apt install` . Click **Copy Command** , close OptiGein Design, then paste the command into a terminal and run it.

apt replaces the old version. Your license, settings and designs stay where they are. You can also download the new package from *My License* on geinforce.com and install it the same way as the first time: `sudo apt install ./optigein-design_<version>_amd64.deb`.

If an update does not go through, see Troubleshooting.

Next: activate your copy

Installing puts the program on your machine. Before you can build designs with it, you enter your license once. That is covered next in License and activation.

Chapter 3

License and activation

OptiGein Design is licensed per device. You activate it once on each computer, and from then on it runs from the license stored on that computer, with or without an internet connection, until the license period ends. This page explains what you need, what activation does, and how to read the messages you might see.

What you need before you start

You need an internet connection for the activation itself, and one of two things: the keys from your confirmation email, or the email address and password of your Geinforce account. Either one activates the same license.

When you buy OptiGein Design, Geinforce sends you a confirmation email with two keys. To activate with keys you need both.

Item	What it looks like	What it is for
License key	`OGDX-YYYY-XXXX-XXXX-XXXX`	Identifies your license. It is safe to write down and to quote to support.
Activation key	A 32 character string	Proves the license is yours. Treat it like a password.

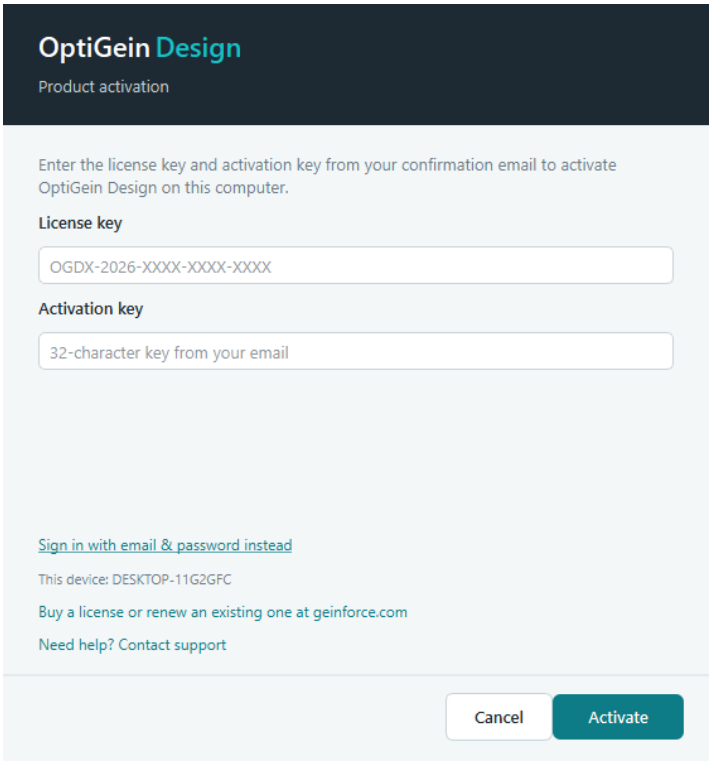
The license key always starts with `OGDX`, then the year, then three more groups of four characters. The activation key is the longer, secret one. Keep the email; you will need both keys again if you move to another computer or reinstall.

> **Keep the activation key private.**
> Anyone who has your license key and activation key can use up your activation slots. The software never shows the activation key back to you after you activate, and support will never ask you to post it in a public place. Store the confirmation email somewhere safe.

Activating on first launch

The first time you open OptiGein Design on a computer, it cannot find a license, so it shows the activation screen before anything else. This is the only screen you will see until activation succeeds; the software does not run unlicensed.

1. Open OptiGein Design. The activation screen appears after the splash.



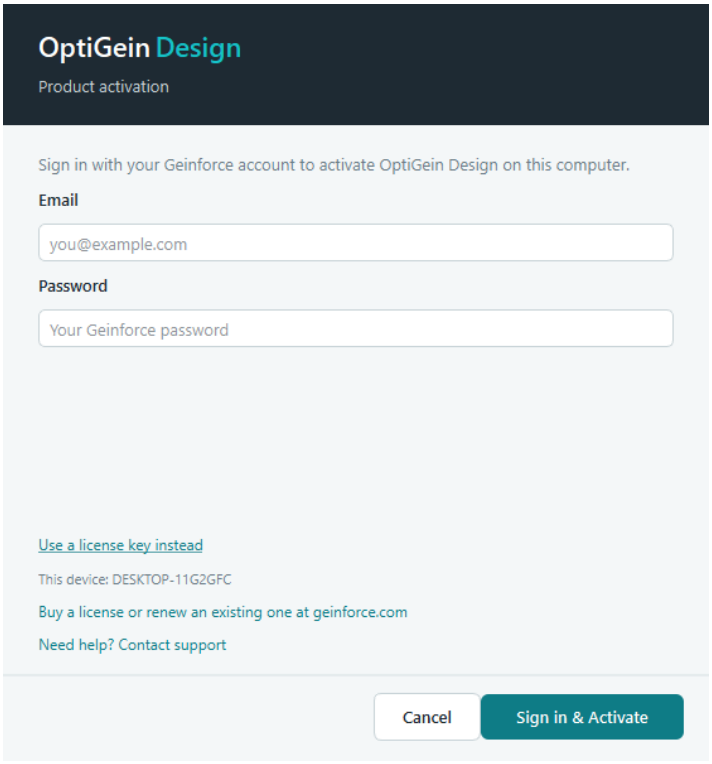
The activation screen. It also shows the name of the computer it is about to bind, a link to sign in with your account instead, and links to buy or renew a license and to contact support.

2. Type or paste your license key into the first field. It is not case sensitive; the software tidies it up for you.
3. Type or paste the 32 character activation key into the second field.
4. Press **Activate** . The software contacts the Geinforce activation server to check the keys and bind this computer.
5. On success you see a short summary: your license type, how many days remain, the date it is valid until, and which activation slot this computer took. The main window then opens on its own.

Signing in instead of typing keys

If you do not have the confirmation email to hand, you can activate with your Geinforce account instead of the two keys.

1. On the activation screen, click **Sign in with email & password instead** , just above the device name. The two key fields are replaced by an email field and a password field.



The same screen after switching to sign in. Use a license key instead takes you back to the key fields.

2. Type the email address and password of your Geinforce account.
3. Press **Sign in & Activate** . The software signs you in, finds the license on your account, and binds this computer to it. You see the same summary as with keys.

What the software stores afterwards is the license, exactly as it does when you activate with keys. You do not sign in again on later launches.

If the connection drops mid check, the software retries a few times on its own before giving up, so a brief network hiccup does not usually need a second attempt from you.

What activation does to the device

Activation binds this specific computer to your license and uses one activation slot. Each license allows a fixed number of devices, shown to you as "device 1 of 3", for example. The software identifies the machine by its operating system installation, works that out once, and keeps it. Restarting, updating the app, or updating the operating system does not consume another slot. Nor does changing how the computer is connected: another Wi-Fi network, a mobile hotspot, a network cable, a docking station or a VPN all leave the activation alone. A genuinely different computer does use a slot.

> **Asked to activate again after changing network?**
> Versions before 1.4.5 included the network adapter in the device identity, so a change of network could make an activated computer look like a new one. Version 1.4.5 fixed this, and an existing activation carries over when you update. See Keeping the software up to date.

Re-activating the same computer with the same license, for instance after a reinstall, does not cost an extra slot. It simply refreshes what is stored locally.

Working offline

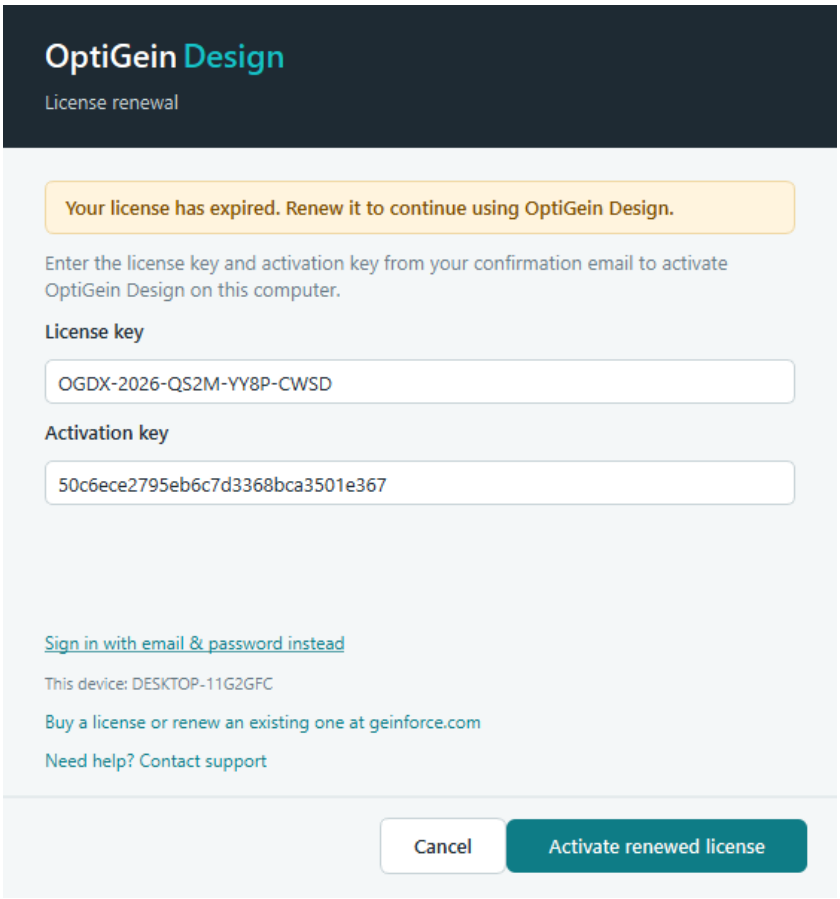
Once a computer is activated, OptiGein Design does not need the internet to start. Each time it opens it reads the license stored on that computer and checks one thing: that the license period has not ended. It does not wait for a server, so the software opens just as quickly on a train or in a laboratory with no network as it does at your desk. This holds for the whole license period, however long the computer stays offline.

When a connection happens to be available, the software quietly refreshes the stored license in the background shortly after the main window opens. That is how a renewal or an extended end date reaches your copy without you doing anything. The refresh never interrupts your work and never uses an activation slot. If there is no connection, nothing happens and nothing is lost.

> **What does need a connection.**
> Activating, activating a renewed license, deactivating a device and checking for updates all talk to the Geinforce server, so they need the internet. Everything else, from building a design to exporting a report, works offline.

When a license expires

If your license has expired, OptiGein Design will not open into the main window. Instead it shows the renewal screen, which is the activation screen relabelled for renewal, with your existing keys already filled in. Renew your license, then activate the renewed license from that same screen.



The renewal screen. Renew your license at geinforce.com, then activate the renewed license here with the same keys, or enter new keys if you were issued a fresh license.

If the server refuses an activation because the license has been revoked or suspended, the message asks you to contact support instead. Renewing on your own does not restore a revoked or suspended license.

A license can also lapse while the software is open. OptiGein Design checks the stored license periodically as it runs, and if the expiry date passes during your session it brings up the same renewal screen. If you close it without renewing, the window closes through the normal path, so you still get the chance to save any unsaved work.

Checking your license status

To see where your license stands at any time, open **Help › License Information**. It reads the license stored on this computer and shows:

Field	Meaning
Status	Active with the number of days remaining, or Expired.
Type	The license type, for example Standard.
Valid until	The expiry date, in your local date format.
Licensed to	The name or email the license was issued to.
Devices in use	How many activation slots are used out of the total, for example 2 of 3.

Moving to a new computer

Activation slots are limited, so free the slot on a machine you are done with before you activate a new one. Choose **Help › Deactivate This Device**. The software tells the server to release this computer's slot, removes the license stored locally, and closes. The next time you start OptiGein Design on that machine it will ask for activation again.

The local license is always cleared, even if the server could not be reached at that moment. If that happens and the slot still shows as used later, contact support and they can clear it for you.

Messages you might see

Most activation problems come with a plain message that tells you what to do. Here are the ones you are most likely to meet.

Message	What it means	What to do
Maximum activations reached	Your license is already active on the highest number of devices it allows, and this is a new one.	On a computer you no longer use, choose Help › Deactivate This Device to free a slot, then activate here. If you cannot reach that computer, contact support and they can release the old device for you.
License has expired	The validity period is over. This is not a connection problem.	Renew your license at geinforce.com, then activate the renewed license on the renewal screen using the same keys.
Your stored license could not be verified	The license stored on this computer is incomplete or cannot be read, so the software cannot rely on it.	Connect to the internet and activate again with the same keys or the same account. If you are online and it still fails, check that your system clock is correct, then contact support.

Message	What it means	What to do
Connection problem, retrying	The activation server did not answer. The software waits a few seconds and tries again on its own.	Wait. If it gives up, check your internet connection and any firewall or proxy, then press Activate again.

Two more can appear when you first type your keys. "License key not recognized" means a typo in the license key, so check it against your email. "Activation key doesn't match this license" means the license key and activation key are not a pair, which usually means one of them was copied wrong.

Keep the computer clock correct.
The end date of your license is compared with the date and time on your computer. A clock that is wrong can make a valid license look expired, or bring up a message that the clock has been set back. Set the date and time correctly and start the software again. The only way to extend a license is to renew it.

Getting help

If a message does not match anything here, or activation keeps failing when you are certain the keys and the connection are fine, contact Geinforce. Have your license key ready. Do not send your activation key by any channel you would not send a password.

Email info@geinforce.com, call +91 9960867592, or visit geinforce.com.

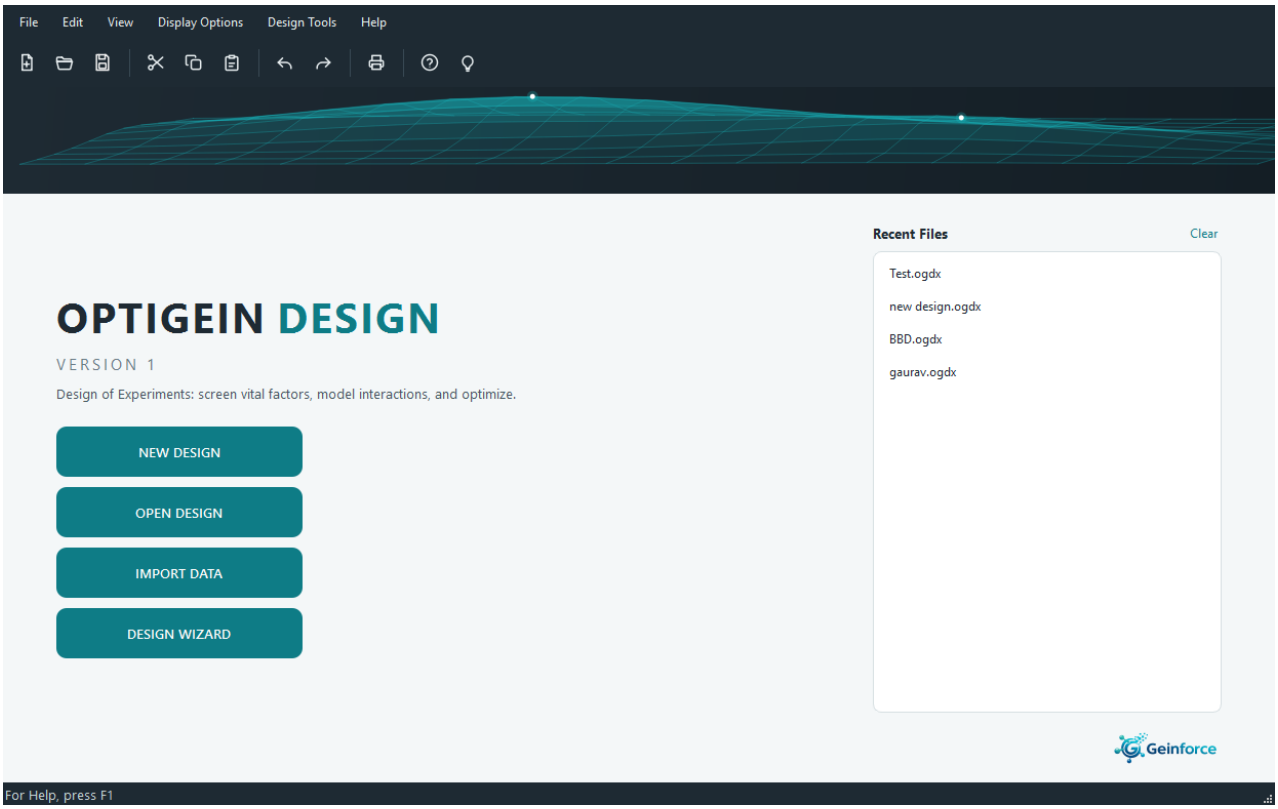
Chapter 4

A tour of the software

Ten minutes, one pass through every screen. You do not need any data yet; the aim is to recognise each part of the software when you meet it later.

The welcome screen

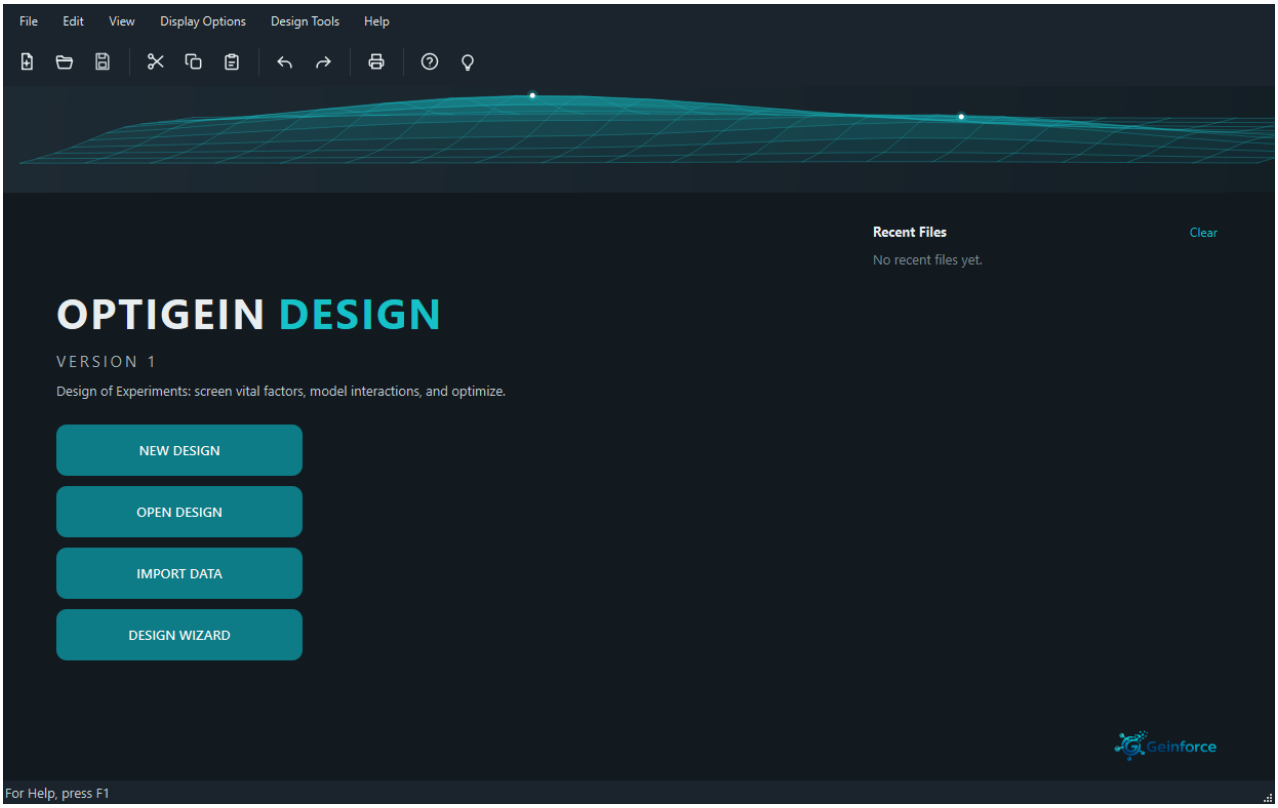
When OptiGein Design opens, it offers you the four things people usually want first: start a new design, open a design you saved earlier, import data you already have, or let the Design Wizard ask a few questions and pick a design for you. Files you worked on recently are listed so you can jump straight back in.



The welcome screen. The menu bar and toolbar above it stay available throughout.

Light and dark

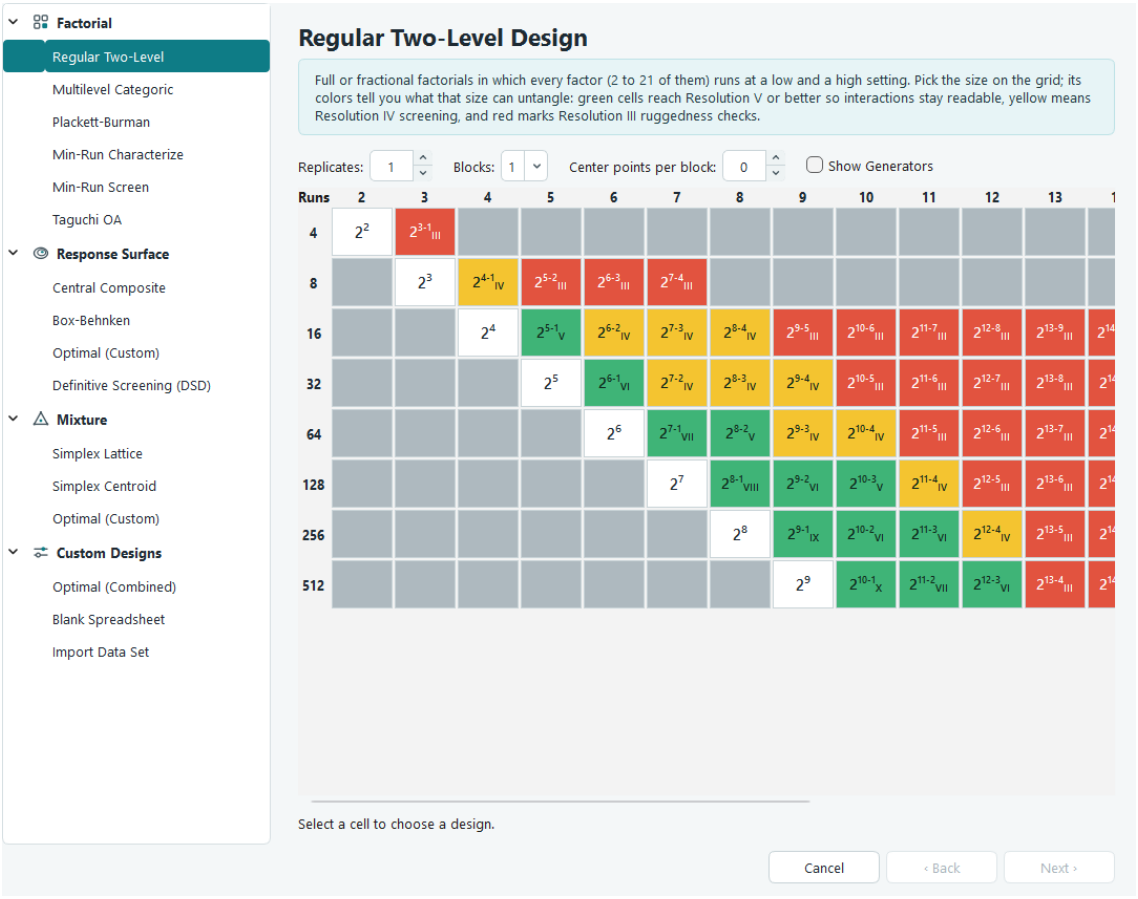
OptiGein Design follows the appearance setting of your operating system. If the computer is set to dark mode, the whole application, including its dialogs, message boxes and tooltips, uses a dark version of the same teal and slate colours. Change the setting in the operating system and the application follows straight away, without a restart. There is nothing to configure inside OptiGein Design itself.



The same welcome screen on a computer set to dark mode. The pictures in the rest of this guide use the light theme; every screen has the same layout in both.

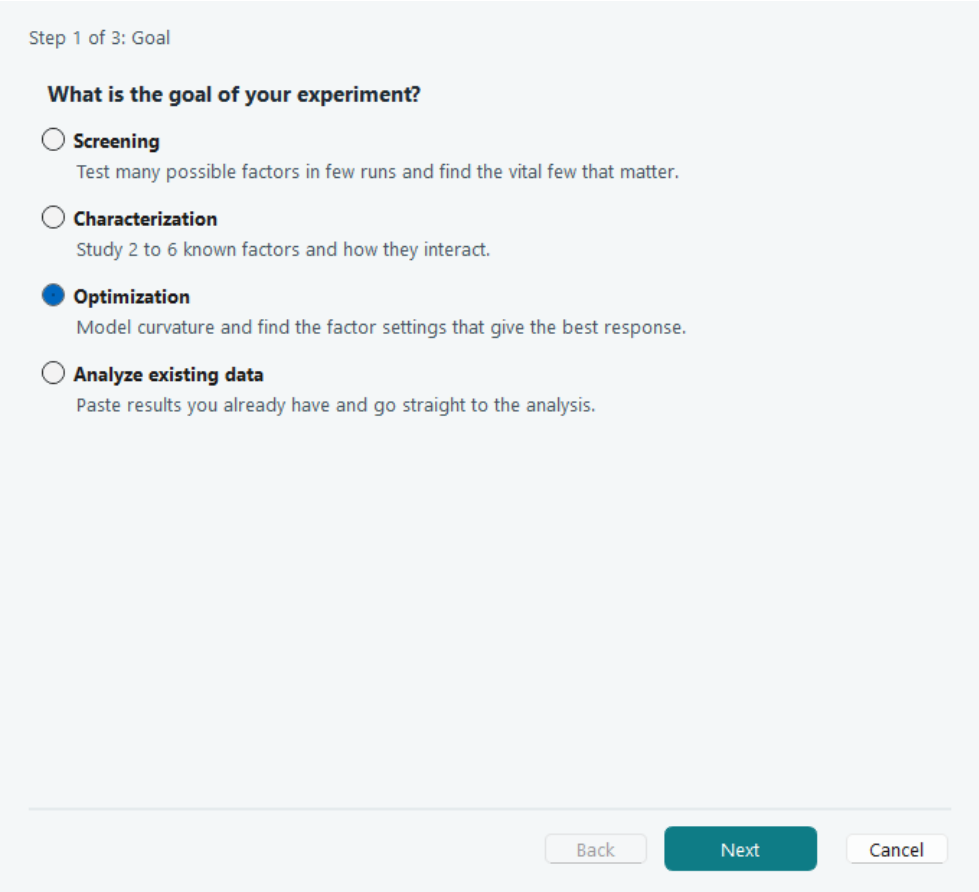
Choosing a design

The New Design screen lists every design the software can build, grouped into four families. Select one on the left and the panel on the right explains what it is for and what it costs in runs. Nothing is built until you press Finish, so you can browse freely.



The design catalogue. Read Which design do I need? if the choice is not obvious.

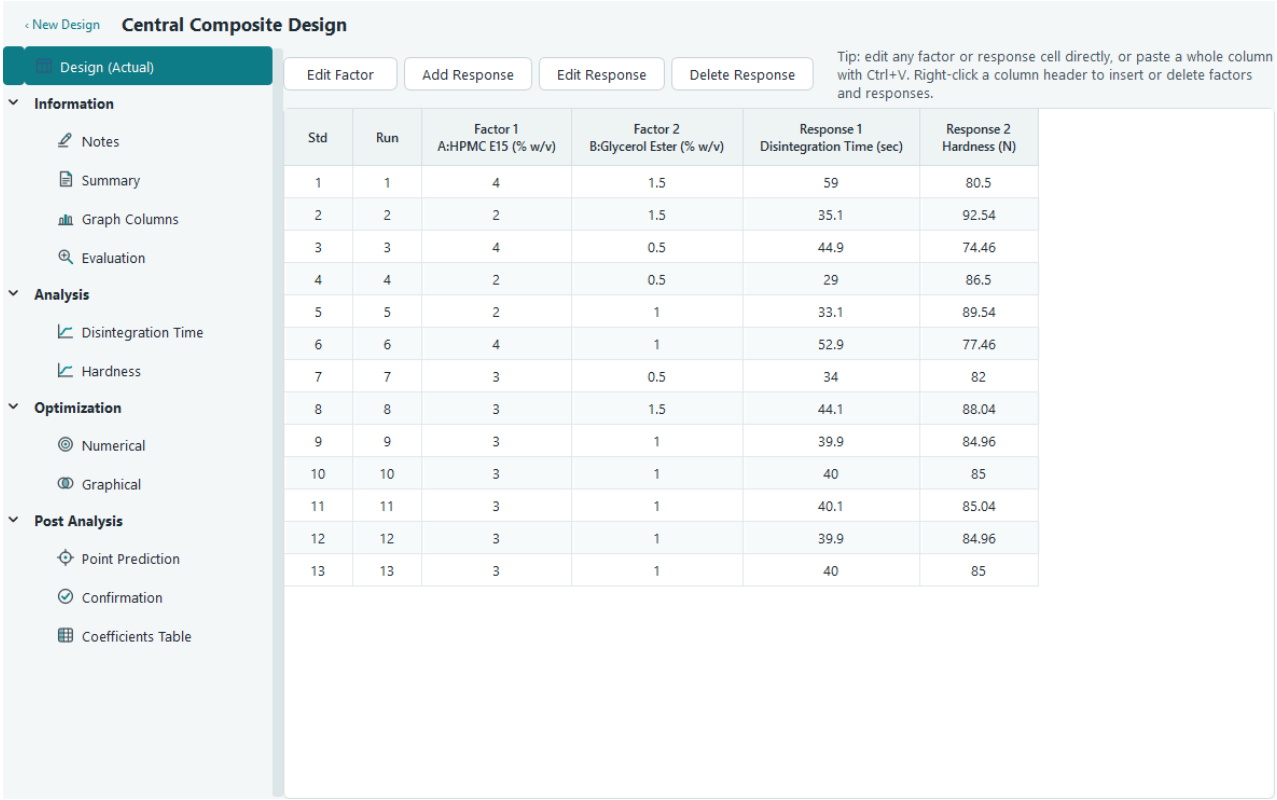
If you would rather answer questions than choose from a list, open the **Design Wizard** from the Design Tools menu. It asks what stage you are at, how many factors you are studying and how many runs you can afford, then recommends a design and builds it.



Step one of the wizard. Screening, characterisation and optimisation are three stages of the same journey, and the wizard explains the difference.

The run sheet

Once a design is built, the run sheet appears. Each row is one experiment. The software has already worked out the settings for you; your job is to perform the runs and type the measured responses into the last columns.



Std	Run	Factor 1 A:HPMC E15 (% w/v)	Factor 2 B:Glycerol Ester (% w/v)	Response 1 Disintegration Time (sec)	Response 2 Hardness (N)
1	1	4	1.5	59	80.5
2	2	2	1.5	35.1	92.54
3	3	4	0.5	44.9	74.46
4	4	2	0.5	29	86.5
5	5	2	1	33.1	89.54
6	6	4	1	52.9	77.46
7	7	3	0.5	34	82
8	8	3	1.5	44.1	88.04
9	9	3	1	39.9	84.96
10	10	3	1	40	85
11	11	3	1	40.1	85.04
12	12	3	1	39.9	84.96
13	13	3	1	40	85

The run sheet for a central composite design on two factors, with two responses to fill in.

Columns are labelled with the factor letter and the name you gave it, so `A:HPMC E15 (% w/v)` is factor A. You can paste a whole column from a spreadsheet with `Ctrl` + `V`, and undo any mistake with `Ctrl` + `Z`.

> **Run order matters.**
> The *Run* column is the order in which you should perform the experiments. It is randomised on purpose, so that anything drifting over time (a warming room, an ageing reagent) does not get mistaken for a factor effect.

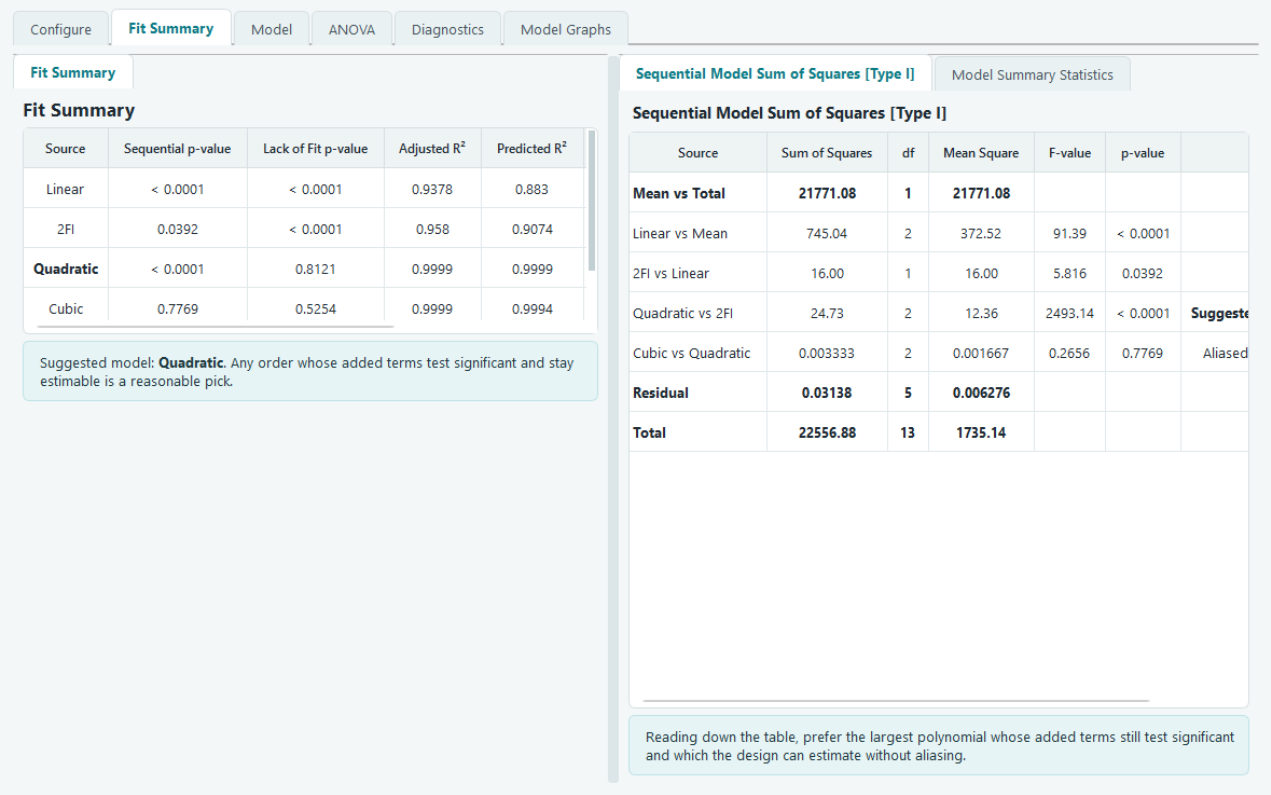
The navigation pane

Everything else lives in the tree on the left. It follows the natural order of an experiment, from the design itself, through analysis, to the optimum.

Branch	What you do there
Design	The run sheet. Enter results, edit factors, add responses.
Information	Notes to yourself, a summary of the design, quick plots of the raw columns, and a formal evaluation of the design before you spend money on runs.
Analysis	One page per response. Fit a model, read the ANOVA, check the residuals, look at the model graphs.
Optimization	Numerical and graphical searches for the settings that satisfy every response at once.
Post Analysis	Predict a response at any settings, plan confirmation runs, and read the coefficients table.

Analysing a response

Click a response under Analysis and you get a strip of tabs that runs left to right in the order you should use them. Configure the model, look at what the data supports, adjust the terms, read the ANOVA, check the diagnostics, then explore the graphs.

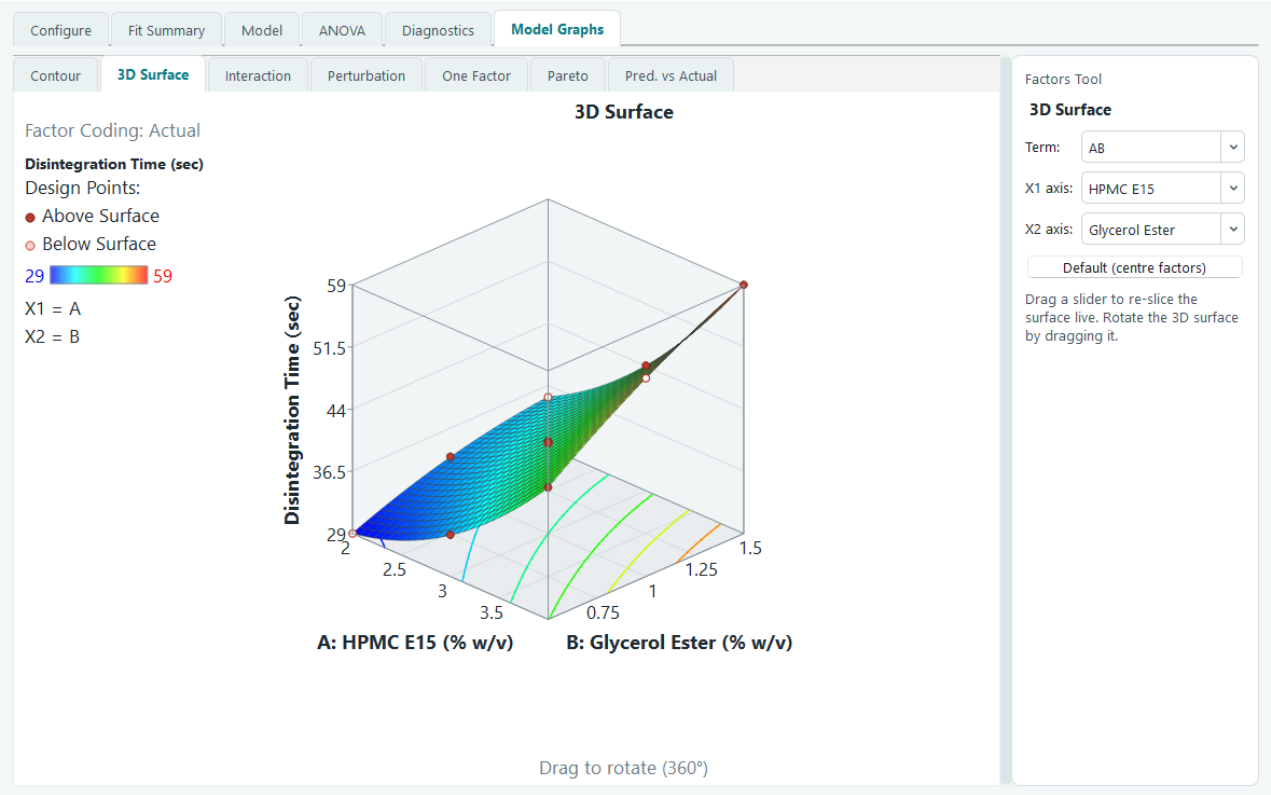


Fit Summary compares linear, two factor interaction, quadratic and cubic models and suggests the one the data supports.

Every table on these screens comes with a short explanation underneath in plain language. If a p-value is small the software says so, and says what that means for the term it belongs to. Full detail is in Fit summary, model and ANOVA.

Seeing the model

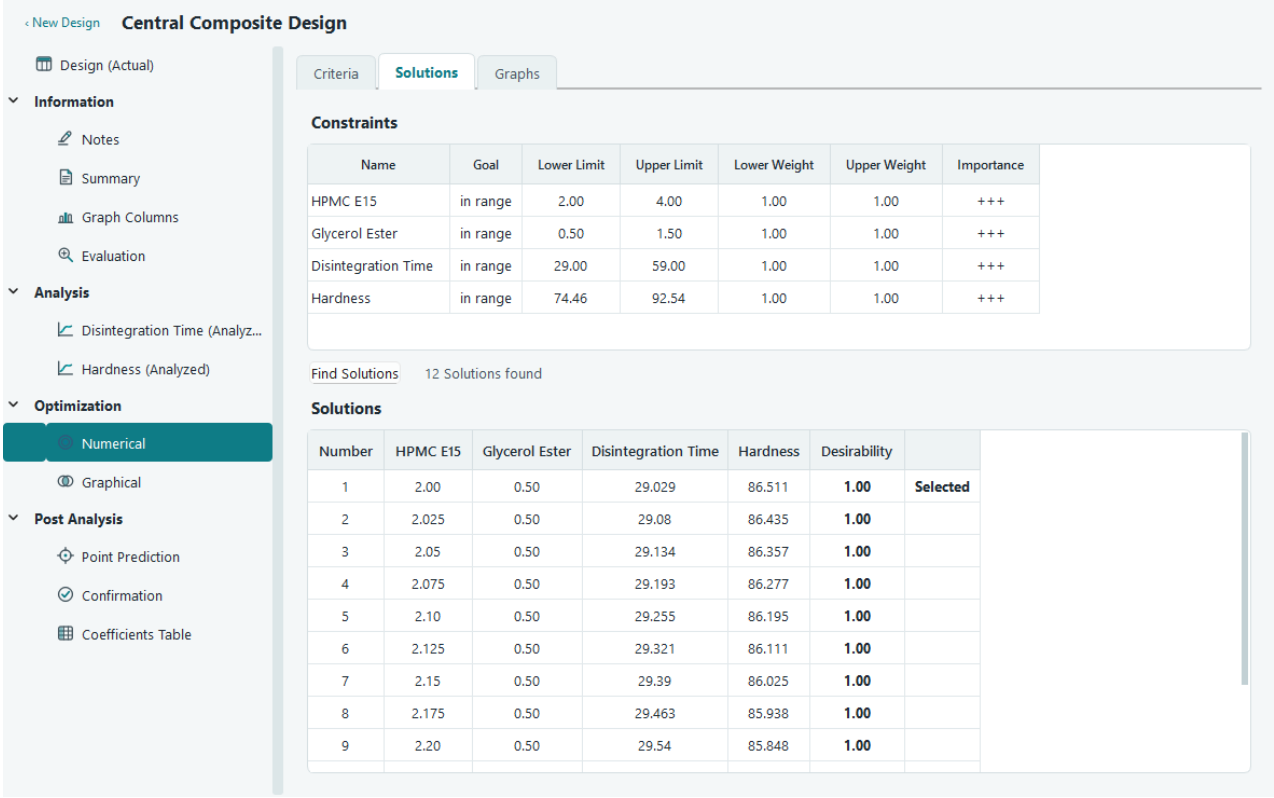
The Model Graphs tab draws the fitted surface. The contour map is the one people put in reports; the 3D surface is the one that makes the shape obvious. Both can be sliced at any setting of the factors you are not plotting.



The fitted surface, with the design points shown above and below it.

Finding the optimum

Under Optimization, give each response a goal. The optimizer combines them into a single desirability score and searches the fitted models for the settings that score highest.



Solutions, best first. Each row is a set of factor settings you could actually run.

What to read next

The fastest way to make all of this concrete is to follow one example from an empty screen to a recommended set of settings.

- Response surface, start to finish , the main tutorial.
- Which design do I need? if you are planning a real experiment now.
- The design workspace for the details of every page in the navigation pane.

Part 2

Choosing a design

Chapter 5

Which design do I need?

OptiGein Design can build sixteen kinds of design. That sounds like a lot, but the choice is usually easy once you know which stage of the experiment you are at. This page walks you through the three stages, points each one at the right family, and lists every design in a single table so you can compare them at a glance.

Three stages, not sixteen decisions

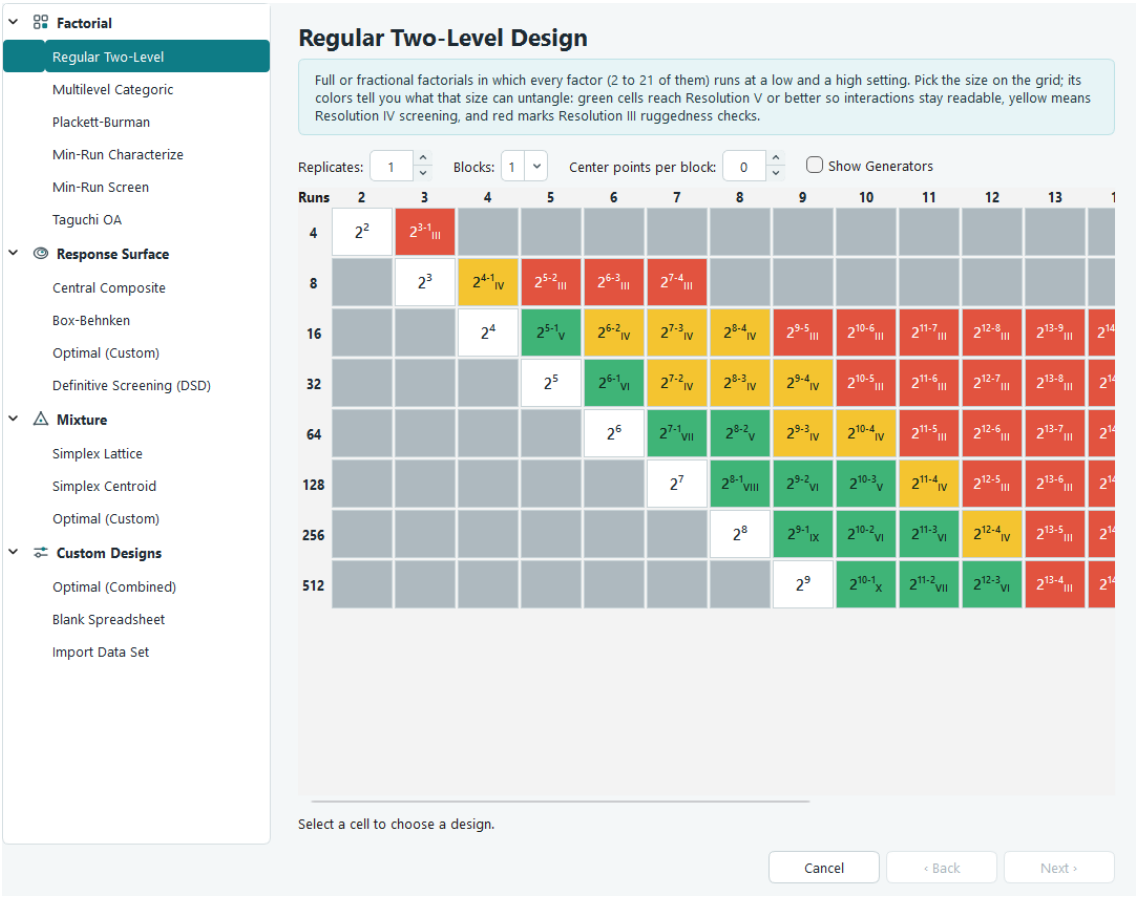
Almost every study moves through the same three stages. You do not always run all three, but knowing where you are is the fastest way to narrow the list.

Screening. You have many candidate factors and no idea yet which ones matter. The goal is to test them all in as few runs as you can afford and find the vital few that move the response. You accept that you cannot separate every interaction at this stage; you just want to know which factors are worth studying further. Screening designs live in the **Factorial** family, with one useful screen (Definitive Screening) in the **Response Surface** family.

Characterisation. You are down to a handful of important factors and now you want to understand how they work together, including which pairs interact. This is what full and fractional two-level designs are built for, so you stay mostly in the **Factorial** family.

Optimisation. You know the factors that matter and you want the settings that give the best result. That means fitting curvature, not just straight-line effects, so each factor needs more than two settings. This is the job of the **Response Surface** family.

The New Design screen groups the catalogue into these families, so once you have named your stage you only have a few entries to read.



The catalogue, grouped into Factorial, Response Surface, Mixture and Custom Designs. Selecting an entry shows what it is for and how many runs it costs.

Mixture designs are a special case

There is a fourth situation that does not fit the stages above. If your factors are the ingredients of a recipe, they are not independent: they have to add up to a fixed total. Push one ingredient up and the others must come down to keep the batch at one hundred percent. An ordinary factorial assumes you can set each factor on its own, so it simply does not apply here. That is why mixtures get their own family.

> **How to tell.**
> If changing one factor forces the others to change so the total stays constant, you have a mixture. Ratios of ingredients, percentages of a formulation and parts of a blend are all mixtures. Use the **Mixture** family. If a recipe and process settings both matter in the same study, use **Optimal (Combined)** under Custom Designs.

The full catalogue

Every design the software builds is below, with its family, how many factors it accepts, what it is for, and a rough run count. Treat the run counts as a guide; the builder shows you the exact number before it creates anything, and options like centre points and replicates change it.

Design	Family	Factors	What it is for	Rough runs
Regular Two-Level	Factorial	2 to 21	Screen or characterise with every factor at a low and a high setting	2 to the power of the factor count for a full set, fewer for a fraction

Design	Family	Factors	What it is for	Rough runs
Multilevel Categoric	Factorial	1 to 12	Cross every named level of every categoric factor	The level counts multiplied together, so it grows quickly
Plackett-Burman	Factorial	2 to 23	Screen many factors for main effects only (Resolution III)	12, 20 or 24
Min-Run Characterize	Factorial	6 to 20	Keep every main effect and every two-factor interaction readable (Resolution V)	1 + k + k(k-1)/2, about 22 at six factors
Min-Run Screen	Factorial	5 to 50	Trim a long factor list, main effects clear of two-factor interactions (Resolution IV)	About two per factor, roughly 2k
Taguchi OA	Factorial	3 to 15	Classic orthogonal arrays for main effects	Fixed by the array, 4 to 27
Central Composite	Response Surface	2 to 10	Map curvature and locate an optimum, five settings per factor	Factorial core + 2k axial + centre points, 13 for two factors
Box-Behnken	Response Surface	3 to 5	Fit a quadratic model with no run at all extremes at once	2k(k-1) + centre points, about 17 at three factors
Optimal (Custom)	Response Surface	1 to 10	Smallest run list for a model you state, numeric and categoric factors welcome	Model terms + 10 by default
Definitive Screening	Response Surface	4 to 20	Compact three-level screen that also shows curvature	About 2k + 1
Simplex Lattice	Mixture	2 to 12	Spread blends evenly across the recipe space to a chosen polynomial degree	Set by component count and lattice degree
Simplex Centroid	Mixture	3 to 10	Every equal-parts blend, best when component synergy is the question	2 to the power of the component count, minus 1, plus axial blends if augmented
Optimal (Custom)	Mixture	2 to 10	Fewest blends that support a Scheffe linear or quadratic model	Model terms + 10 by default
Optimal (Combined)	Custom Designs	2 to 8	One experiment for the recipe and the process together	Crossed model terms + 10 by default
Blank Spreadsheet	Custom Designs	1 to 20	An empty run sheet you fill in by hand or by paste	However many rows you set

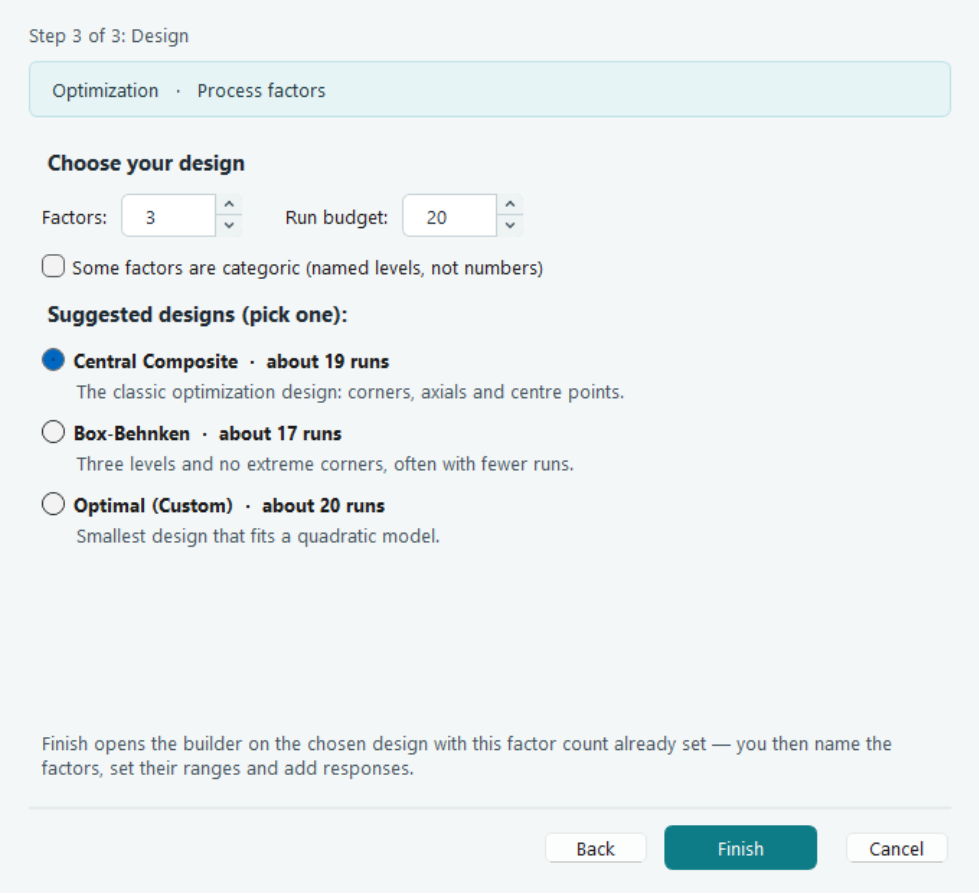
Design	Family	Factors	What it is for	Rough runs
Import Data Set	Custom Designs	1 to 30	Bring in results you already have and go straight to analysis	Matches the rows you paste

In the run counts, *k* is the number of factors. The worked example used throughout this documentation is a face-centred central composite design on two numeric factors, A: HPMC E15 (2 to 4 % w/v) and B: Glycerol Ester (0.5 to 1.5 % w/v). Its run count follows the Central Composite row: four factorial corners, four axial runs and five centre points give 13 runs in all.

Read the panel.
Click any entry in the catalogue and the panel on the right tells you what that design does and, once you set the factor count and options, exactly how many runs it will cost. Nothing is built until you press Finish, so you can compare a few candidates freely before you commit.

Let the wizard choose for you

If you would rather answer a few questions than read the whole catalogue, open the **Design Wizard** from the Design Tools menu. It asks three things: what you are trying to do (screen, characterise, optimise, or just analyse data you already have), whether your factors are independent process settings or mixture components, and how many factors and runs you have in mind. It then lists a short set of suitable designs with an estimated run count for each, and marks any that would exceed your run budget.



The wizard's last step. Set your factor count and run budget, then pick from the suggestions. Finish opens the builder on that design with the factor count already filled in.

The wizard is a good starting point when the choice is not obvious, or when you want a quick sanity check on how many runs an idea will cost. When you know exactly what you want, going straight to the catalogue is faster.

Go deeper on a family

Once you have a family in mind, its own page explains every option and what it does to the run sheet.

- Factorial designs , for screening and characterisation.
- Response surface designs , for mapping curvature and optimising.
- Mixture designs , for recipes and formulations.
- Custom designs and import , for combined studies, blank sheets and existing data.

Chapter 6

Factorial designs

Factorial designs answer the first question of any experiment: out of everything you could change, which factors actually move the result, and by how much. They run each factor at a small number of settings and vary them together, so a handful of runs tells you about many factors at once. This page covers the six designs in the Factorial family and the ideas you need to pick between them.

You reach all six from the New Design screen, listed under **Factorial** in the catalogue on the left. Select one and the panel on the right explains what it does and how many runs it costs before you build anything. If you are not sure which family you need at all, start with Which design do I need?

Resolution, and why the two-level grid is coloured

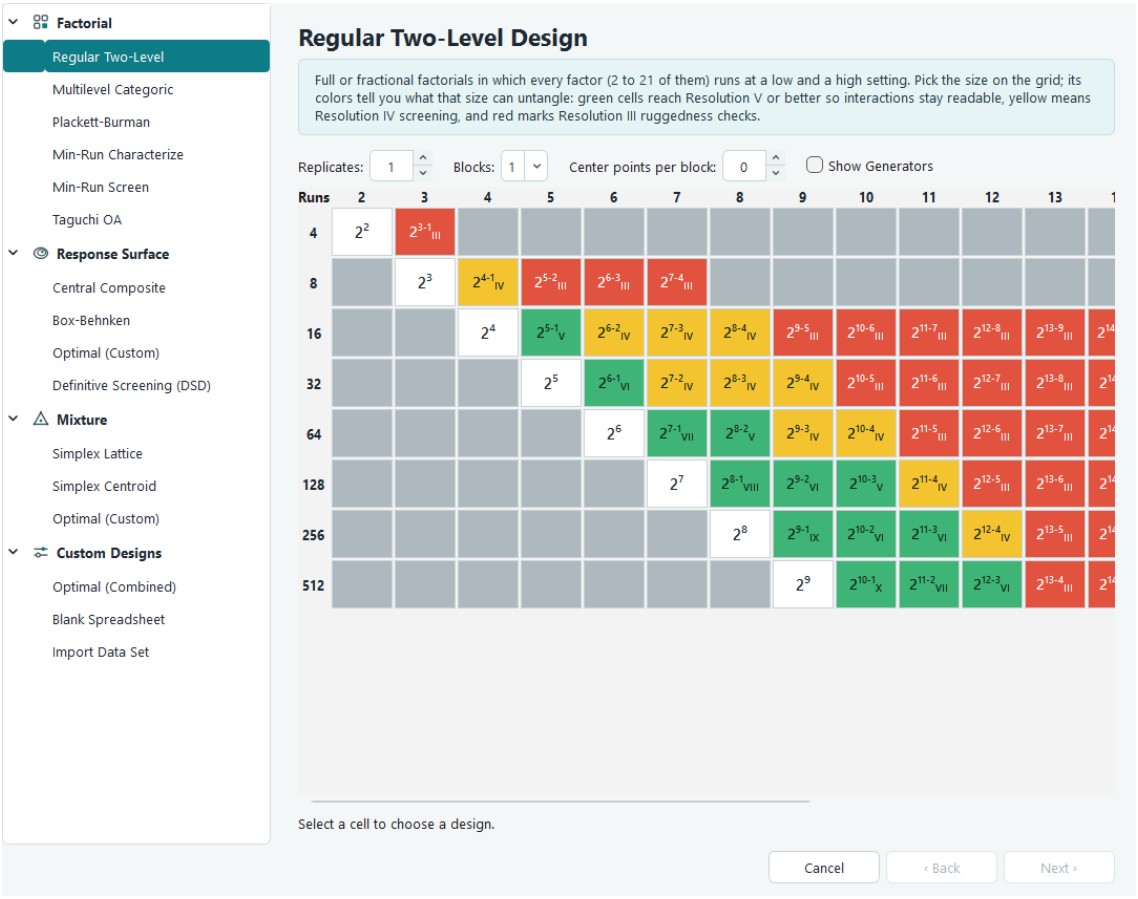
When you cannot afford to run every combination of factors, you run a fraction of them. The price of that saving is *aliasing*: some effects can no longer be told apart, because they were measured on the same set of runs. Resolution is the shorthand for how bad the aliasing is. A higher Roman numeral means cleaner effects. The three you will meet are these.

Resolution	What stays clear	What is tangled	Use it for
III	Main effects.	A main effect is aliased with two-factor interactions, so a big reading could be the factor or an interaction you cannot separate.	A quick ruggedness or sorting check when you are willing to assume interactions are small.
IV	Main effects are clear of every two-factor interaction.	Two-factor interactions are aliased with each other, so you learn that an interaction exists but not which pair caused it.	Screening a long list down to the few factors that matter.
V	Main effects and every two-factor interaction are all clear of one another.	Interactions only overlap three-factor and higher terms, which are usually negligible.	Characterising a short list of factors, interactions included.

OptiGein Design paints this straight onto the two-level grid so you can see the trade at a glance. Green cells reach Resolution V or better, yellow cells are Resolution IV, and red cells are Resolution III. Full factorials sit in the leftmost column with no colour fill, because they alias nothing at all.

Regular Two-Level

This is the workhorse of the family. Every factor, from 2 up to 21 of them, runs at a low and a high setting. You choose the size on a grid whose rows are the number of factors and whose columns are the fraction, from the full factorial on the left through progressively smaller fractions to the right. Each cell shows the run count and its resolution colour, so you are choosing runs and aliasing together in one click.



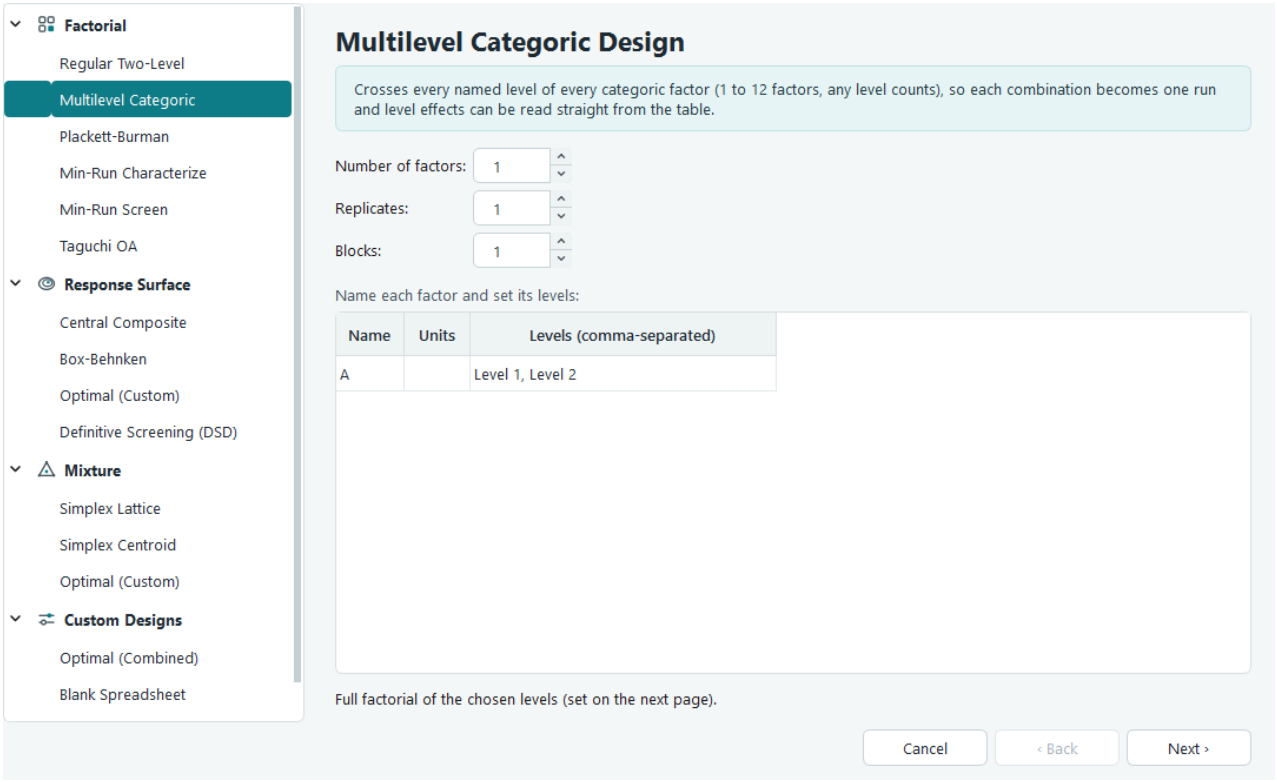
The two-level grid. Green is Resolution V or better, yellow is IV, red is III; the full factorials on the left have no aliasing.

Below the grid are four controls. **Replicates** repeats the whole design so you can estimate pure error. **Blocks** (1, 2, 4 or 8) splits the runs into groups when you cannot do them all under identical conditions, for example across two days. **Center points per block** adds runs at the midpoint of every factor. **Show Generators** reveals the defining relation (written `I = ...`) for a fractional design, which is the exact statement of what is aliased with what.

> **Read the aliases before you commit.**
> A green full or fractional two-level design is often the right first step before a response surface study like the tablet example used elsewhere in this guide. Confirm which of your factors matter here, cheaply, then spend the larger run budget only on those.

Multilevel Categoric

Use this when a factor is a category rather than a dial, or when two settings are not enough. It takes 1 to 12 factors, each with its own list of levels (a supplier, a grade, three temperatures), entered as a comma-separated list. The design then crosses every level of every factor, so each combination becomes exactly one run and the effect of each level can be read straight from the averages in the run sheet.

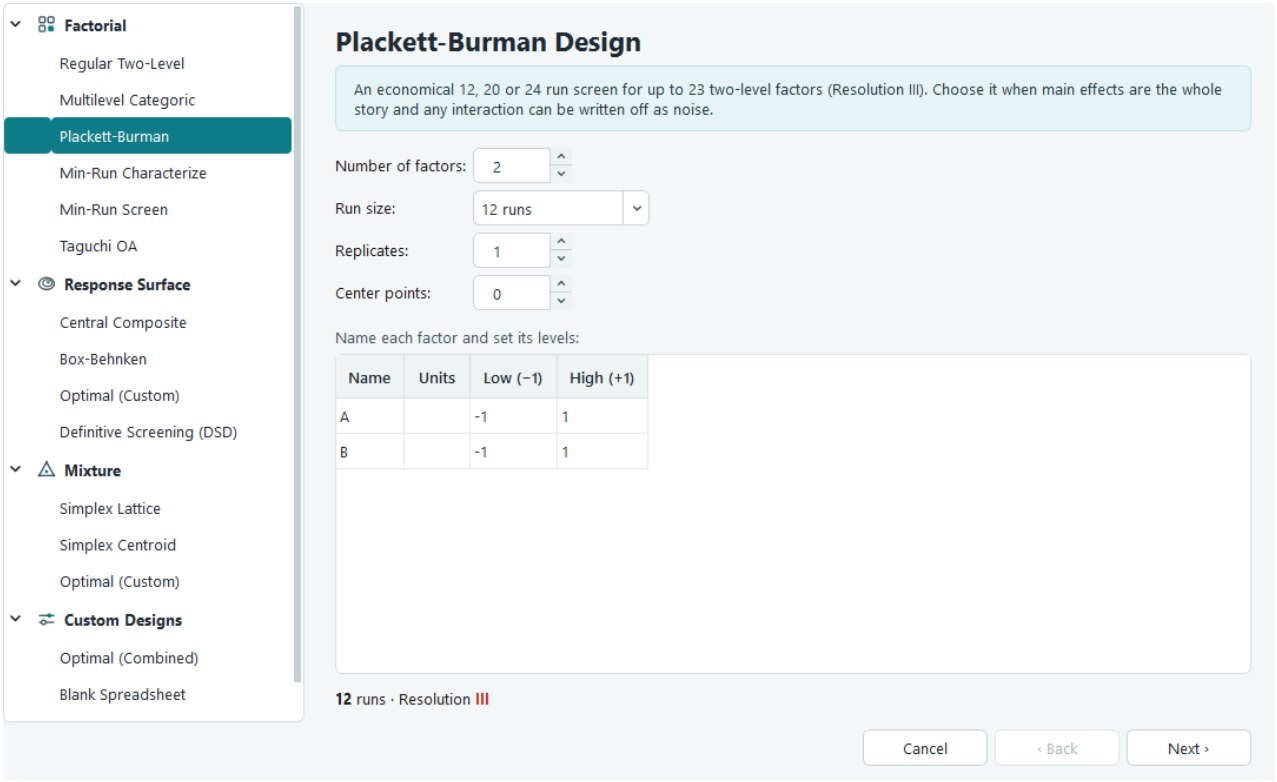


Multilevel Categoric crosses every level of every factor. Set the number of factors, replicates and blocks, then name the levels.

The run count is the product of the level counts, so it grows quickly: three factors at three levels each is already 27 runs before any replication. Replicates and Blocks work as they do for the two-level design. Because this is a full crossing, there is no aliasing to worry about, but the size is the limit rather than the resolution.

Plackett-Burman

When you have many two-level factors and you only want to know which ones matter, Plackett-Burman is the cheapest honest answer. It offers 12, 20 or 24 run sizes and handles up to 23 factors, which is why the number of factors you may enter is capped at one below the run size you pick. Every design is Resolution III.

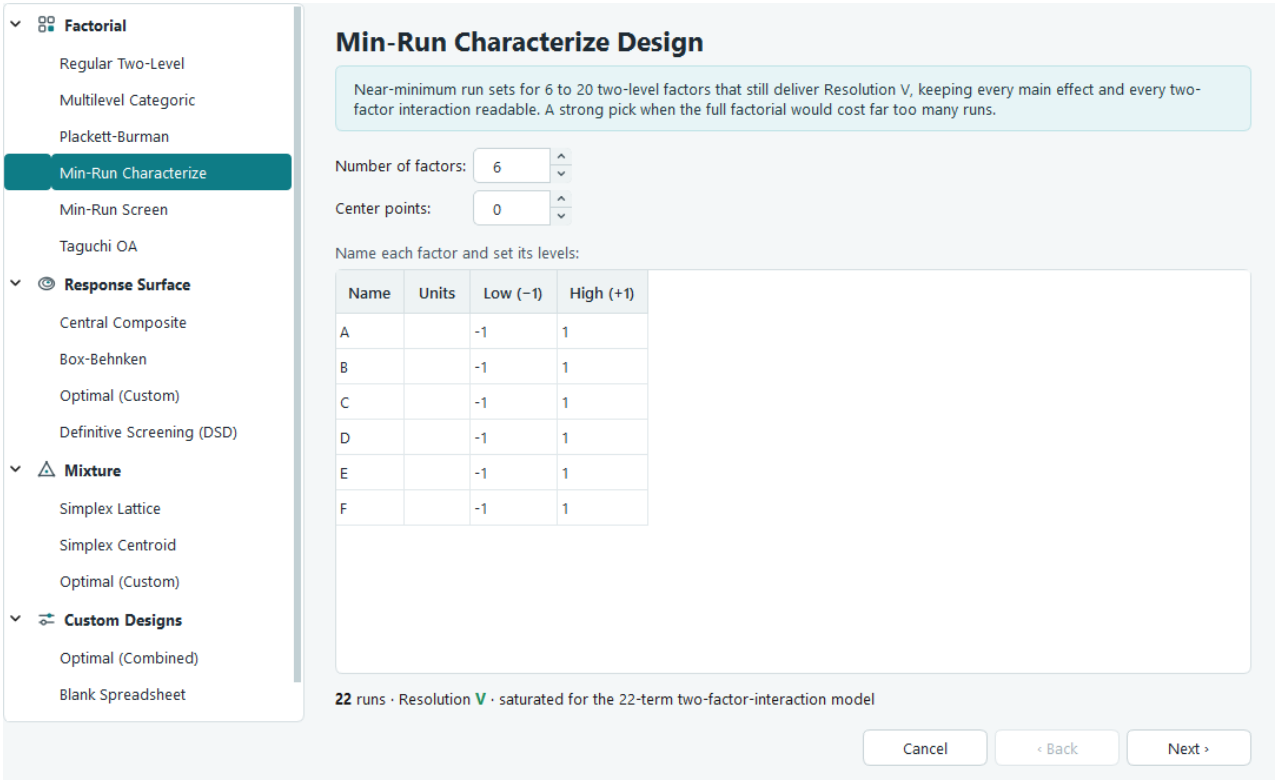


Plackett-Burman. Choose the run size, then the number of factors; you can add replicates and center points.

The catch is baked into that Resolution III: main effects are partially aliased with two-factor interactions. You can conclude that a factor looks important, but if an interaction is really driving the response, Plackett-Burman will not tell you. Treat it as a sieve, not a final answer, and follow a promising result with a higher-resolution design on the survivors.

Min-Run Characterize (Resolution V)

A full Resolution V two-level design for many factors can run into hundreds of experiments. This design gives you the same clean reading of every main effect and every two-factor interaction for 6 to 20 factors, at close to the fewest runs the model allows. The software builds it by D-optimal coordinate exchange rather than from a textbook fraction, so it is not tied to a power-of-two size.

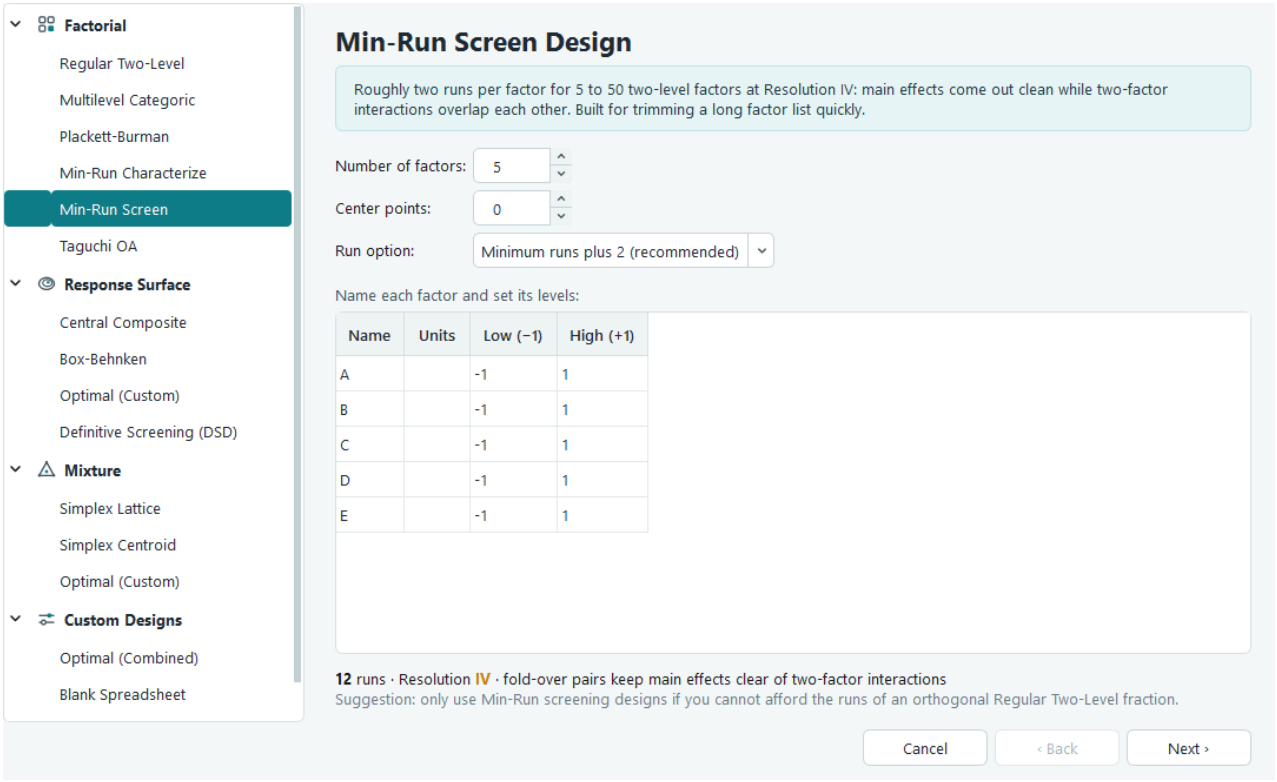


Min-Run Characterize. The preview reports the run count for the full two-factor interaction model at Resolution V.

The run count equals the number of terms in the two-factor interaction model, which is `1 + k + k(k-1)/2` for `k` factors. That makes the plan saturated: at the minimum it has no spare runs to estimate error. This is why the **Center points** control matters here. Adding a few gives you replication for a pure-error estimate and a curvature check. Choose this design when you need interactions but a full Resolution V fraction is too expensive.

Min-Run Screen (Resolution IV)

This is the leanest screening design in the family, for 5 to 50 factors at roughly two runs per factor. It is built as a fold-over pair: a compact main-effects half plus its mirror image, which guarantees Resolution IV. Main effects come out clear of every two-factor interaction, while the two-factor interactions overlap one another.

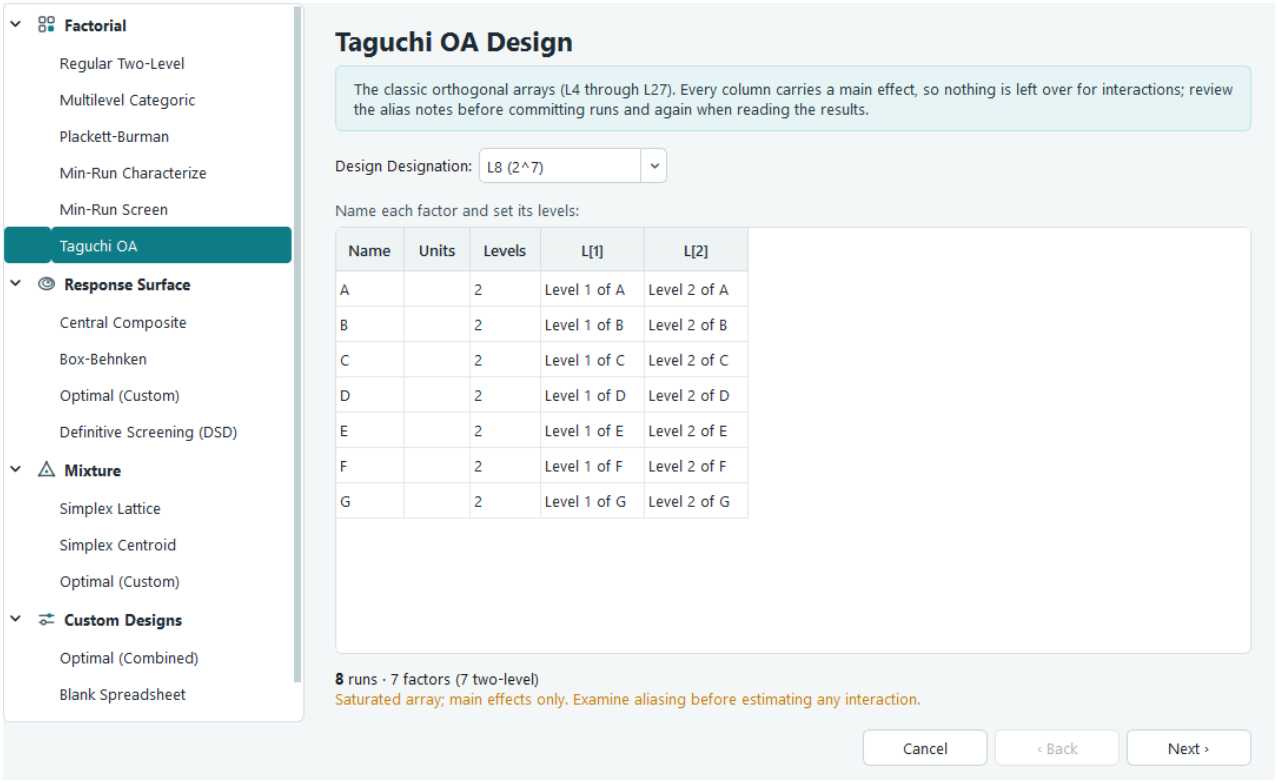


Min-Run Screen. The Run option chooses between the bare minimum and minimum runs plus two.

The **Run option** offers minimum runs, or minimum runs plus 2. The plus-2 choice is the default because it lifts the fit away from saturation and gives a steadier estimate. You can conclude which factors have real main effects and that some interactions are present, but not which interactions. The software itself suggests you prefer an orthogonal Regular Two-Level fraction whenever you can afford one, and only reach for this design when you cannot.

Taguchi OA

The Taguchi orthogonal arrays are a fixed catalogue of balanced tables: L4, L8, L9, L12, L16, L18, L25 and L27. You pick the designation and the array decides the number of factors and their levels for you; the columns range from two-level (for example L8 holds seven two-level factors) to three and five levels. You only name the factors and their levels. Every column is balanced against every other, so main effects are estimated with equal precision.



Taguchi OA. The designation you pick fixes the number of columns and their levels; you fill in names.

A Taguchi array leaves no room for interactions.
These arrays are saturated: every column carries a main effect, so nothing is left over to estimate a two-factor interaction. Any interaction that is really present will fold into the main-effect columns and quietly bias them. Read the alias notes on the design before you commit runs, and again when you read the results. If interactions are part of the question, choose a two-level design at Resolution IV or V instead.

Center points, replication and run order

Three settings show up across several of these designs and are worth understanding once.

Center points are runs with every numeric factor set to its midpoint. They do two jobs. Repeated, they give you a pure-error estimate that does not depend on the model being correct. Compared against the average of the factorial runs, they reveal curvature: if the centre sits well off the line between the corners, the response is bending. A two-level design can detect that curvature but cannot map it, so a strong signal is your cue to move up to a response surface design.

Replicates repeat runs so that random measurement scatter can be separated from real effects. More replication sharpens every estimate and is the honest way to earn degrees of freedom for the error term, at the cost of more runs.

Run order is randomised for a reason.
Every design here fills in a randomised *Run* column and leaves the *Randomize run order* box ticked by default. Perform the experiments in that order. It stops anything that drifts over time, a warming room or an ageing reagent, from lining up with a factor and masquerading as its effect. Untick it only when the run order is genuinely fixed for you.

Chapter 7

Response surface designs

These designs are for the stage after screening, when you already know which factors matter and you want the shape of the response, not just its direction. They let you fit curvature, find a peak or a valley, and settle on the best combination of settings.

Why you need more than two levels

A two-level design measures each factor at a low and a high setting only. Two points define a straight line, so a two-level design can tell you that a response rises or falls, but it cannot tell you that the response bends. Real processes bend all the time: a response climbs, flattens at an optimum, then falls again. To see that shape you have to test each factor at three or more settings, and you have to fit a model with squared terms. A model with squared terms is a response surface, and the four designs on this page are the ways OptiGein Design builds one.

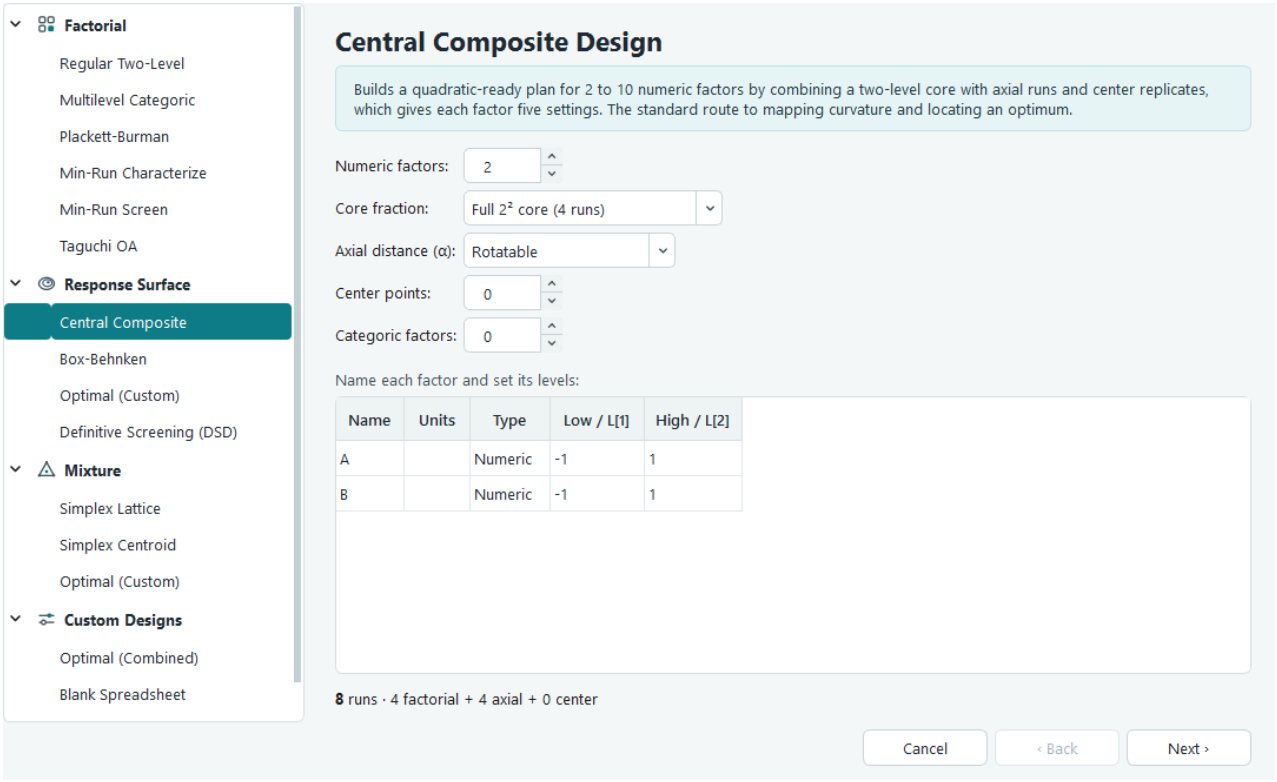
What a quadratic model buys you.
A quadratic model has a term for every factor, every pair of factors, and the square of every factor. The squares are what let the surface curve, which is what lets the optimizer find an interior best point rather than pushing every factor to a corner.

Central composite

The central composite design is the usual first choice for two to ten numeric factors. It is built in three parts, and each part does a job:

- **A factorial core.** A two-level full or fractional factorial on the corners of the design space. This part estimates the main effects and the two-factor interactions, exactly as a screening design would.
- **Axial points, also called star points.** Two extra runs per factor, placed out along each factor axis with the other factors held at their centre. These runs are what make the squared terms estimable, so they are the reason the surface can curve.
- **Centre replicates.** Several runs with every factor at its middle setting. Repeating the centre gives a direct measure of pure experimental error and shows whether the middle of the surface sits where a flat model would predict.

Between the corners, the axial points and the centre, each factor ends up at five distinct settings, which is plenty to fit and check a quadratic model.



The central composite configuration page. Set the numeric factors, the core fraction, the axial distance and the number of centre points, then name each factor and its low and high levels.

The axial distance, alpha

Alpha is how far out the star points sit, measured in coded units where the corners are at plus or minus one. The value of alpha decides whether those runs stay inside the factor ranges you typed or reach beyond them. The configuration page offers three choices.

Choice	Axial distance	What it means physically
Rotatable	The fourth root of the number of factorial points, about 1.41 for two factors.	Prediction is equally precise in every direction at a given distance from the centre. The star points fall outside the low-to-high range you set, so each factor is run a little beyond its stated limits.
Spherical	The square root of the number of factors.	Every non-centre run sits on one sphere around the centre point. The star points again reach outside your ranges.
Face-centered ($\alpha = 1$)	Exactly one.	The star points land on the faces of the cube, at the low or high level itself. No run ever leaves the ranges you set. Choose this when a setting past your limits is impossible, unsafe or too expensive to try.

When a factor cannot go beyond its range.
If a factor is already at a hard limit at its high level, a purity that cannot exceed a ceiling or a pump at its maximum speed, use the face-centred choice. Rotatable and spherical both ask for runs outside the range, which you would not be able to perform.

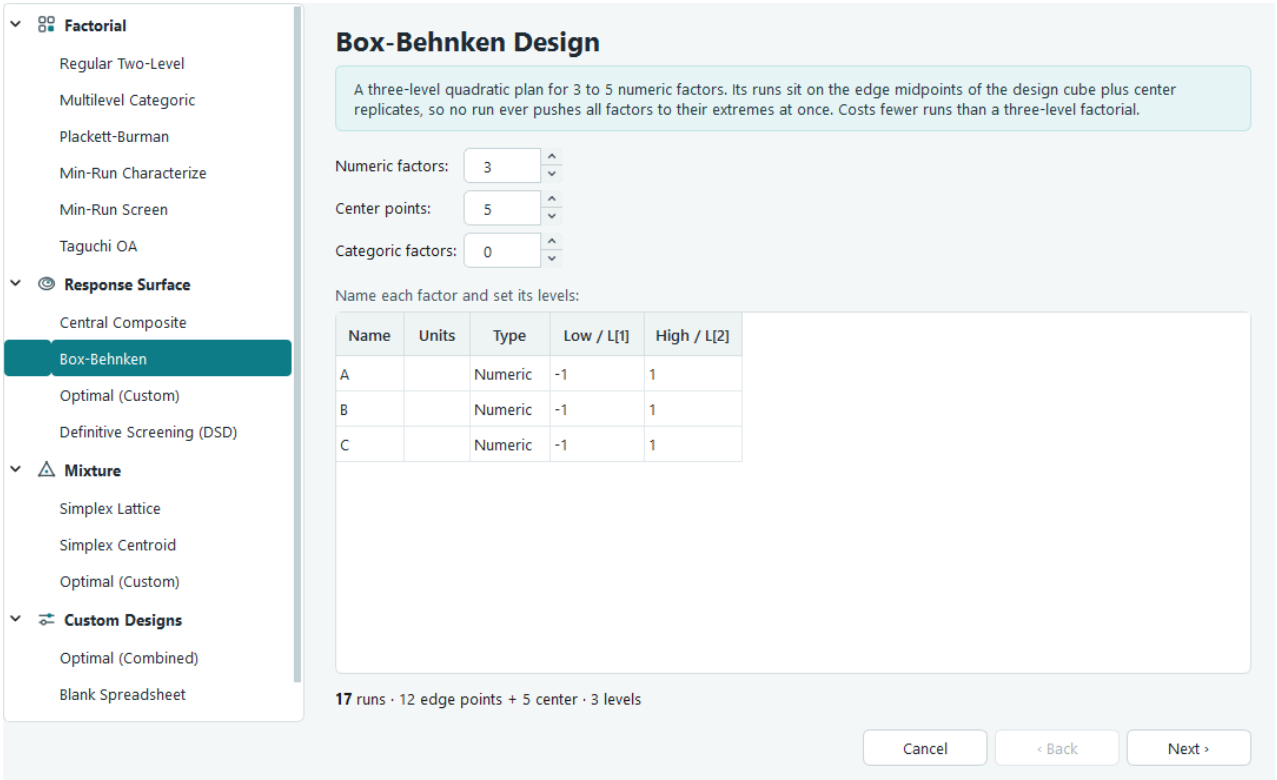
The worked example in this documentation is a face-centred central composite on two numeric factors, A: HPMC E15 from 2 to 4 % w/v and B: Glycerol Ester from 0.5 to 1.5 % w/v. It has 13 runs: four factorial corners, four axial points and five centre replicates. Because it is face-centred, every run stays inside 2 to 4 % w/v for HPMC and 0.5 to 1.5 % w/v for the ester, which is why you never see an out-of-range value on the run sheet.

The **Core fraction** field only matters once you have enough factors that a full factorial core would be wasteful. With many factors you can pick a fraction of the core, which trims the corner runs while keeping the axial and centre runs intact. With two or three factors the core is simply full.

Box-Behnken

Box-Behnken is a leaner three-level design for three to five numeric factors. Instead of corners and star points, its runs sit at the midpoints of the edges of the design cube, with the remaining factors held at their centre. Each numeric factor takes just three settings: low, middle and high.

The useful property is that no single run puts every factor at an extreme at the same time. Every run has at most two factors at a limit while the rest sit in the middle. If your corner combinations are the risky ones, all factors maxed together, a Box-Behnken design avoids them entirely. It also costs fewer runs than a full three-level factorial would.



The Box-Behnken configuration page. It is available for three to five numeric factors and starts with five centre points.

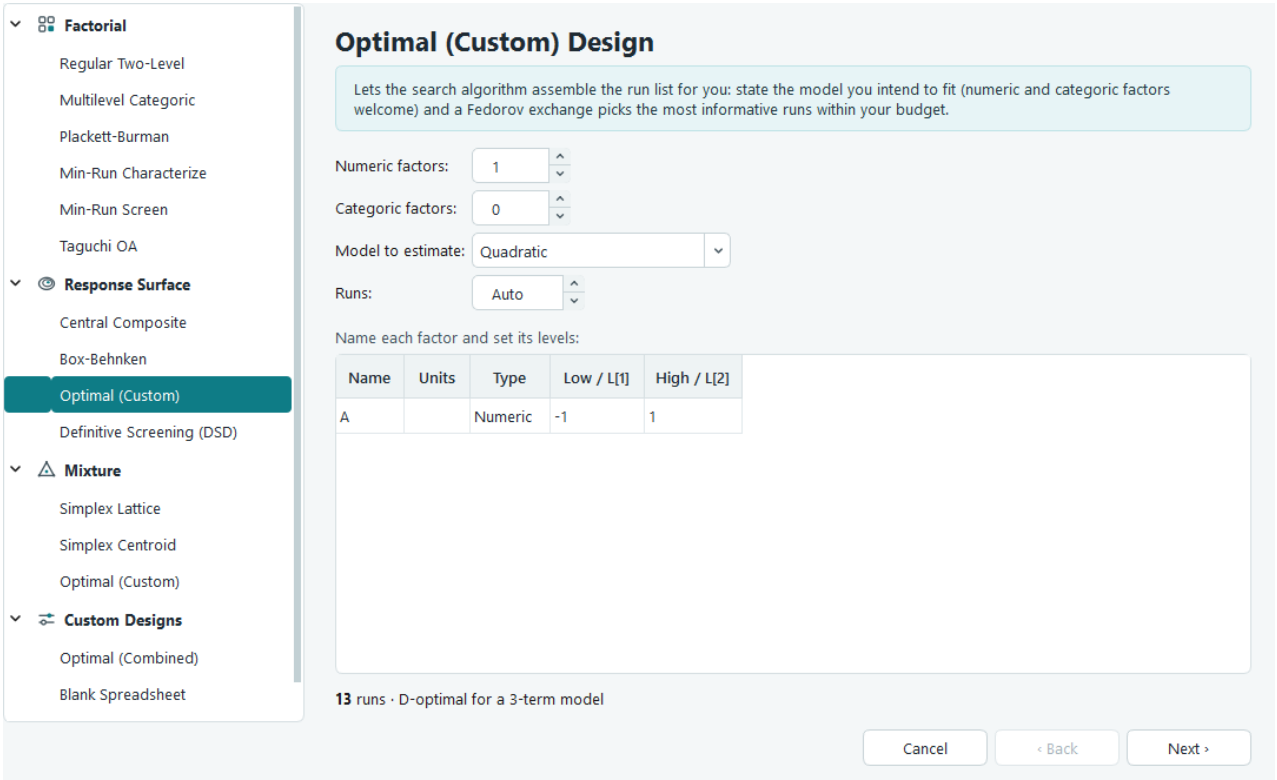
The number of edge runs grows with the number of factors: each pair of factors contributes four runs, so the total is four times the number of factor pairs. That works out to twelve runs for three factors, twenty-four for four and forty for five. On top of that, the design starts with five centre replicates by default, so a three-factor Box-Behnken is seventeen runs. You can change the centre count on the configuration page.

Central composite or Box-Behnken?
Both fit the same quadratic model. Reach for central composite when you can run settings a little past your limits and want the extra corner and star coverage. Reach for Box-Behnken when three levels per factor is enough and you want to keep every run away from the all-extremes corners.

Optimal (Custom)

The two designs above have a fixed geometry. An optimal design does not: you tell the software what model you intend to fit and how many runs you can afford, and it chooses the runs for you. This is the design to use when your run budget is set in advance, when you mix numeric and categoric factors, or when the standard shapes do not fit your constraints.

On the configuration page you set the numeric factors, any two-level categoric factors, the **model to estimate** (main effects, two-factor interactions or quadratic) and the number of **runs**. Leave the run count on Auto and the software allocates the number of terms in your model plus ten, which leaves room to test for lack of fit and to estimate error. A Fedorov exchange algorithm then searches for the set of runs that measures your chosen model most precisely, swapping candidate runs in and out until the information it gathers stops improving.



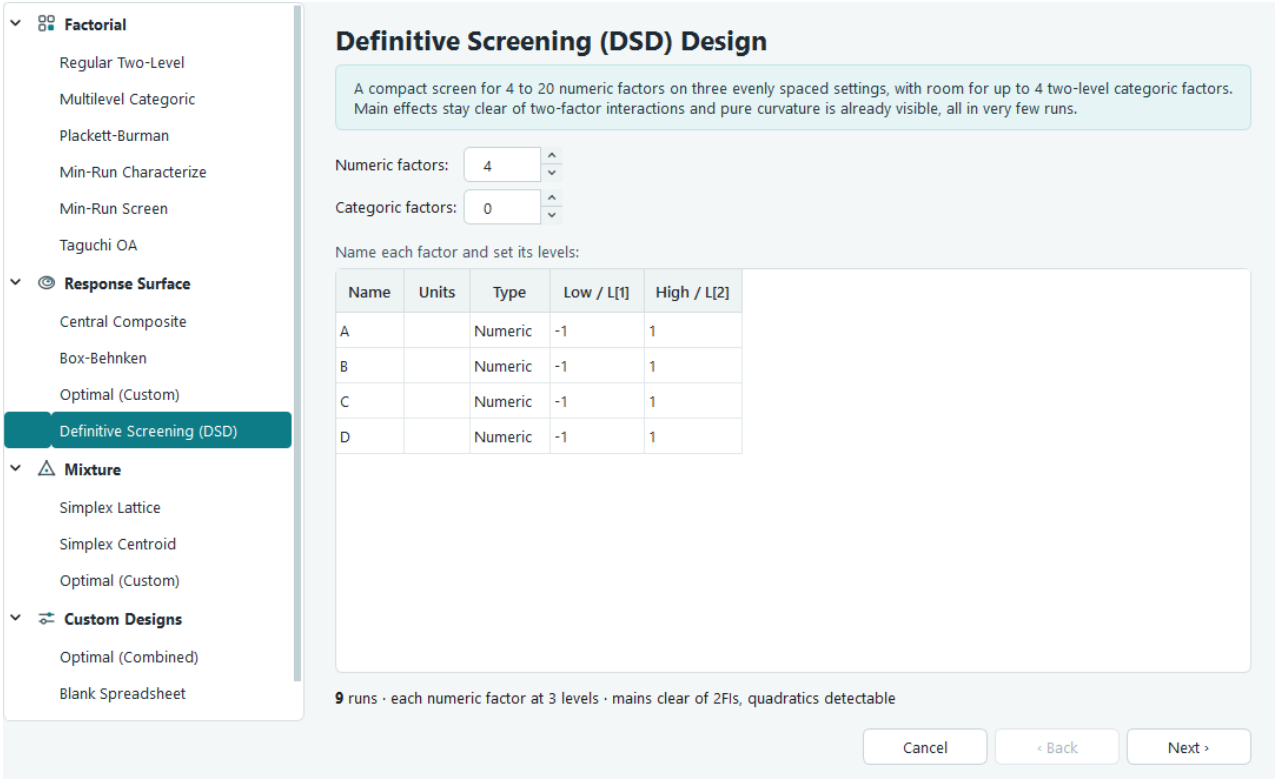
An optimal response surface design. State the model and the run budget, and the exchange algorithm assembles the run list.

Give the model enough runs.
A quadratic model needs at least as many runs as it has terms just to be estimable, and more than that to be checked. If you ask for fewer runs than the model needs, the configuration page tells you the minimum. Add a few beyond that minimum so you can still test the fit.

Definitive screening

A definitive screening design, or DSD, sits between screening and response surface work. It is built for four to twenty numeric factors, each set at three evenly spaced levels, with room for up to four two-level categoric factors alongside them. Its appeal is how much it separates in very few runs, a little over twice the number of factors.

Two properties make it worth knowing. First, the main effects stay clear of the two-factor interactions, so a large main effect is never an interaction in disguise. Second, because every numeric factor runs at three levels, pure curvature is already visible, meaning you can see whether a factor bends the response before you commit to a full quadratic design. If only a few factors turn out to matter, a DSD often carries you far enough to model them without a second experiment.



A definitive screening design. Main effects stay clear of two-factor interactions and curvature shows up, all in a compact run sheet.

What to read next

- Response surface, start to finish walks the face-centred central composite example from an empty screen to a recommended set of settings.
- Evaluating a design shows how to check a response surface design before you spend money on the runs.
- Which design do I need? compares the families side by side if you are still deciding.

Chapter 8

Mixture designs

Reach for a mixture design when the factors are the parts of one recipe and the response depends on the proportions, not the amounts. This page explains why a recipe needs its own kind of design and covers the three mixture types the software builds: Simplex Lattice, Simplex Centroid and Optimal (Custom).

What makes a mixture different

In an ordinary experiment you can set each factor on its own. Temperature does not care what the pressure is. A recipe is not like that. Its components are proportions of one whole, so they are tied together by a single rule: they have to add up to a fixed total. If a tablet blend is filler, binder and lubricant, and you raise the binder, something else has to come down to make room. You can never move one component while holding the rest still, because the total is not allowed to change.

That one constraint changes everything about the design. The response you care about is usually the blend, not the batch size. A film at 30% plasticiser behaves the same whether you mix one gram or one kilogram, so what the experiment varies is the set of proportions, and those proportions always sum to one (or to whatever batch total you set).

> **The total is a knob.**
> You set the mixture total on the design screen. Leave it at 1 to work in plain proportions, or set it to a real batch size so the run sheet reads in grams or millilitres. Either way the components in each run add up to that total.

Why a factorial does not work here

Picture three components on three ordinary axes, each running from 0 to 1. A factorial or a response surface design would put points at the corners of a cube: all-low, all-high, and every mix in between. Almost none of those corners is a legal recipe. The point where all three components are at their maximum does not exist, because three things at 100% each cannot share one batch. The only points that obey the sum rule lie on a flat slice cutting across that cube.

That slice is the **simplex**, and it is where every mixture design lives. Its shape depends on how many components you have:

Components	Shape of the design space	The corners are
2	A line segment	All of A, or all of B
3	A triangle	The three pure components
4	A tetrahedron	The four pure components
q	A q-cornered simplex	Each single pure component

The corners of the simplex are the pure components. The edges are two-part blends. The interior holds the blends that use every component at once. A mixture design is simply a sensible set of points spread across this simplex, and the three types below spread them in different ways.

How a mixture model is fitted

The sum constraint also reshapes the model. An ordinary polynomial starts with an intercept, a plain constant sitting under everything else. In a mixture that constant cannot be separated from the component terms: because the components always add to the same total, any constant could be hidden inside the linear terms and nobody could tell the difference. The intercept and the component terms are aliased with one another.

There are two equivalent ways to resolve this. The classical Scheffe form drops the intercept and keeps one linear coefficient for every component. OptiGein Design uses the other one, known as the slack variable form: it keeps the intercept and drops whichever single component term the sum constraint has made redundant. That term is reported as *Aliased* on the Fit Summary and Model tabs. This is the software telling you the design cannot estimate it separately, not a sign that something has gone wrong.

Both forms predict identically.
Every fitted value, every line of the ANOVA table, every contour and every optimisation result is the same under either parameterisation. What changes is how you read the coefficients.

Because the intercept is kept, a coefficient in the OptiGein equation is *not* the response you would expect from that pure component, which is how a Scheffe linear coefficient is usually read. Read the coefficients the way you read them everywhere else in the software: as the shift in the response as that component increases, with the dropped component taking up the balance. If you want the value for a pure component, do not do the algebra by hand. Use Point Prediction, set that component to its maximum and the others to their minimums, and read the number off directly.

Squared terms such as A^2 do not appear in a mixture model. The sum rule lets you rewrite any squared term using the linear and cross product terms, so it carries no new information and is left out. What remains are the component terms and one cross product term for each pair of components.

Mixture model	Terms	Count for q components
Linear	One per component, one of which is aliased away	q
Quadratic	Components plus every pair	$q + q(q-1)/2$

Setting a minimum and a maximum for a component

Real recipes rarely allow a component to run from nothing to everything. The design screen lets you give each component a Low and a High, and the software respects them.

A **minimum** simply shrinks the blend region. If a preservative must always be at least 2 percent, every blend the software builds carries at least 2 percent of it, and the remaining 98 percent is what the design explores. The runs still add up to the total, and each component can still reach as high as the others allow: with everything else held at its own minimum, a component reaches its own minimum plus whatever is left over.

A **maximum** is different. If a maximum actually bites, meaning the component could otherwise have reached higher, it slices a corner off the blend region and leaves a shape that a lattice or a centroid cannot cover with its regular pattern of points. Rather than quietly building blends that break the limit you typed, Simplex Lattice and Simplex Centroid stop and tell you. The **Optimal (Custom)** mixture design has no such restriction, because it chooses its runs from a pool of candidate blends and simply ignores the ones that break a bound.

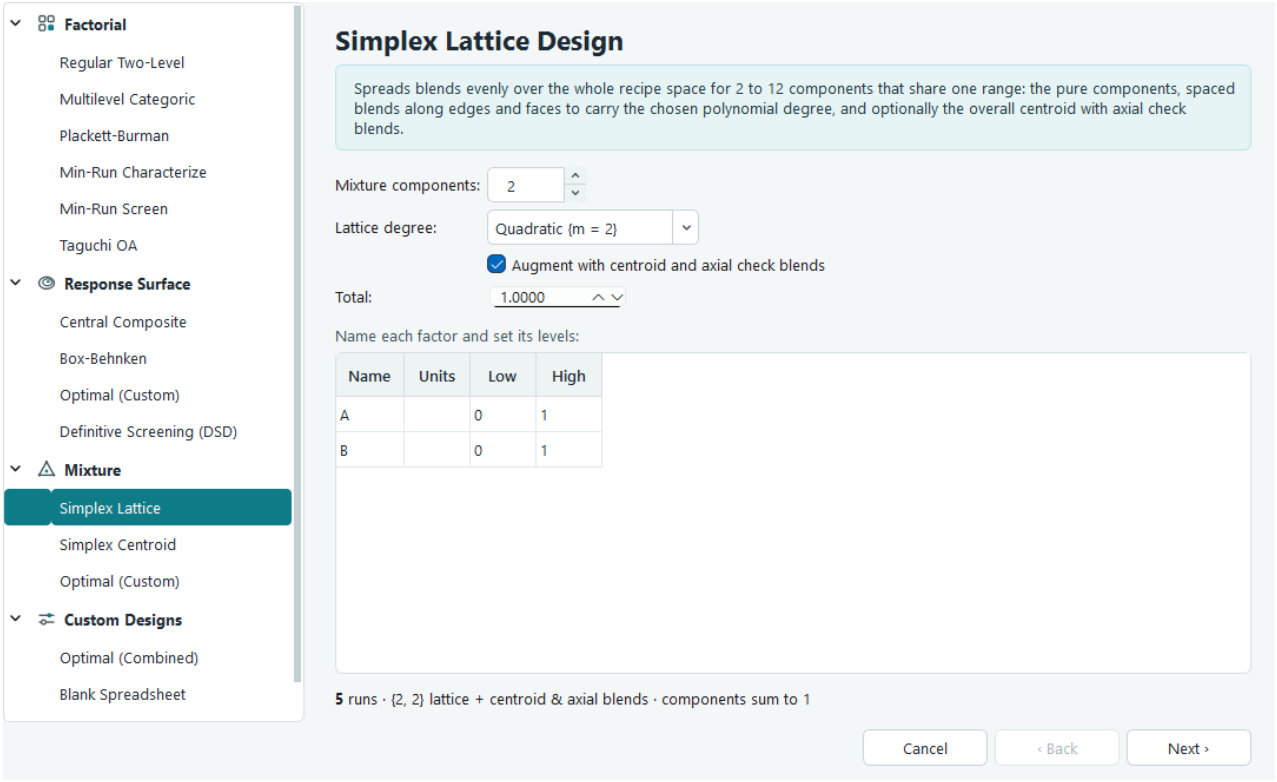
If a maximum is refused.
Either raise that maximum until it no longer bites, or build an Optimal (Custom) mixture design, which honours both bounds.

The cross-product coefficients are the useful part. A positive coefficient on the A×B term means the two components together do better than either one alone, a synergy. A negative one means they work against each other. That is exactly the question a formulator wants a mixture experiment to answer.

You read the model on the analysis screen.
The mixture-specific model is fitted for you once you enter the responses. See Fit summary, model and ANOVA for how to move from a linear blend to a quadratic one and check whether the extra pairwise terms earn their place.

Simplex Lattice

A simplex lattice spreads points evenly over the whole simplex on a regular grid. You choose the number of components, from 2 to 12, and a lattice degree written as m. The degree sets the spacing: each component is placed at the proportions 0, 1/m, 2/m and so on up to 1, and the design uses every combination of those levels that still adds up to the total. A higher degree samples the edges and faces more finely and supports a higher-order blending model, so match m to the model you intend to fit: 1 for a linear blend, 2 for a quadratic one, and higher if you expect a more curved response.



Simplex Lattice in the design catalogue. The panel on the right shows the run count for the components and degree you pick.

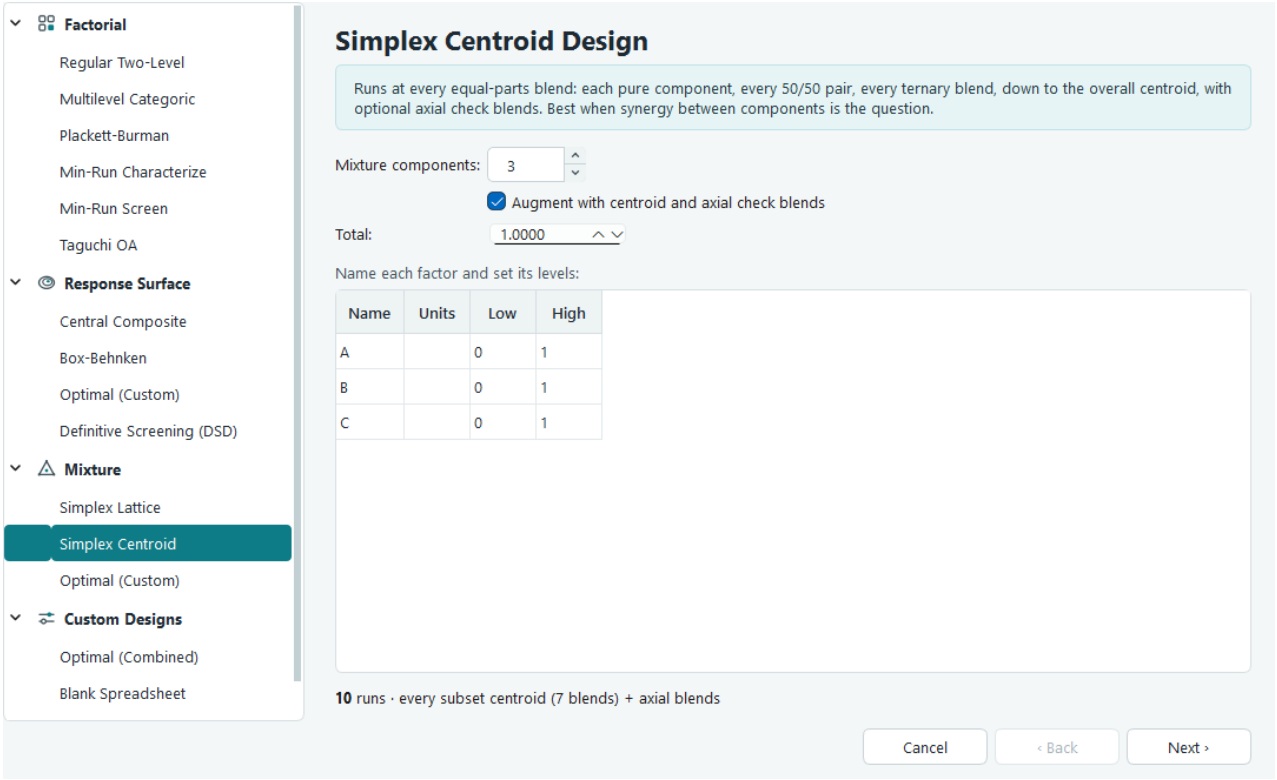
For three components at degree 2 that gives six runs: the three pure components at the corners and the three fifty-fifty blends at the edge midpoints. The number of grid points grows with both the component count and the degree, so the software caps a lattice at 1000 runs and asks you to lower the degree if you ask for more.

A plain lattice keeps all of its points on the outside of the simplex, on the corners, edges and faces. That is fine for fitting the model but it leaves the middle of the space unchecked. The **augment** option, which is on by default, fixes this. It adds the overall centroid, the blend that uses every component in equal parts, and one axial check blend per component sitting partway between the centroid and each corner. Those extra points let you see whether the model that fits the boundary also predicts the interior well, which is where real

formulations usually live.

Simplex Centroid

A simplex centroid is built around equal-parts blends rather than a grid. For q components, from 3 to 10, it runs every non-empty combination of components mixed in equal parts. That means each pure component on its own, every fifty-fifty pair, every equal three-way blend, and so on, all the way down to the single overall centroid that mixes all q components equally. The run count is a clean $2^q–1$, so three components give seven runs and four give fifteen.



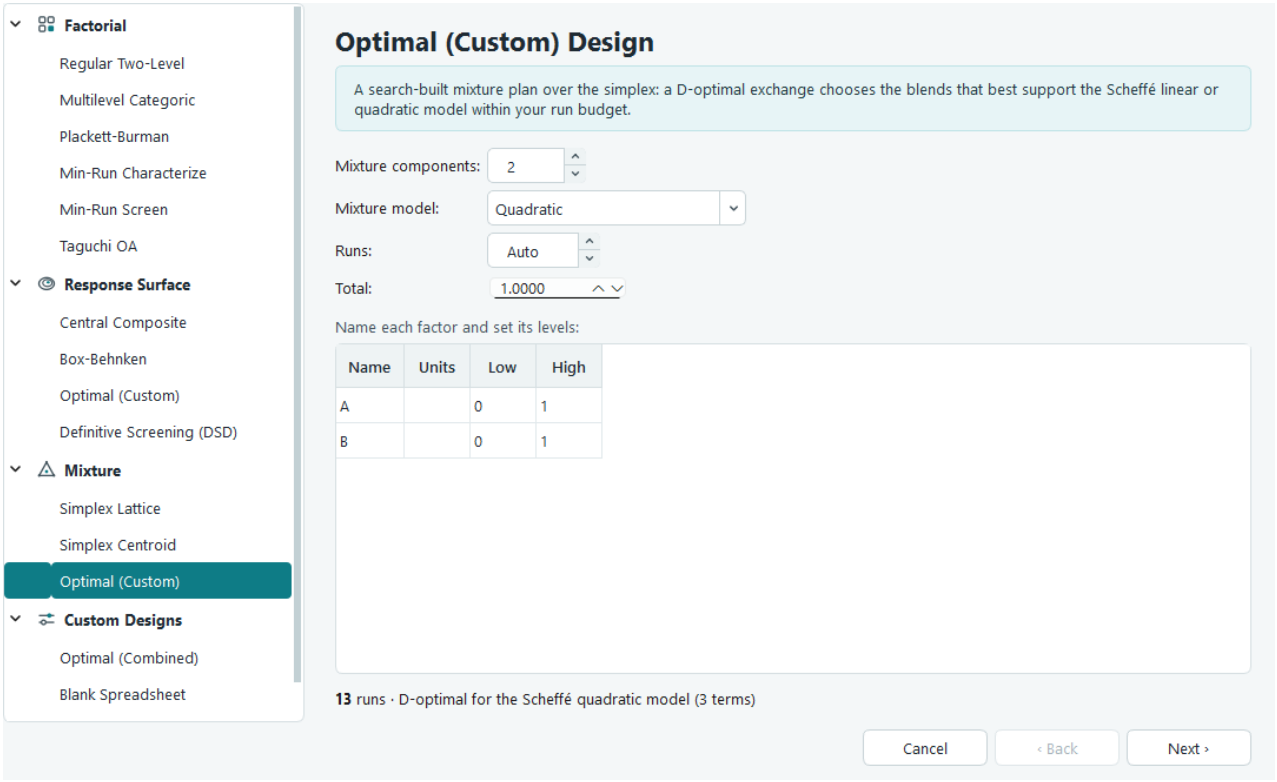
Simplex Centroid in the catalogue. Every subset of components appears once, blended in equal parts.

Because it visits blends of every size, the centroid design is the natural choice when the question is about interaction between components: does a three-way blend behave like the average of its pairs, or does something new happen when all three are present at once. Like the lattice, it offers the same augment option, which adds one axial check blend per component so the interior of the simplex is not left to the model alone.

The run count doubles with each component.
A ten-component centroid is 1023 base runs before augmenting. If you have many components and only a linear or quadratic blend to fit, an Optimal (Custom) mixture will get you there in far fewer runs.

Optimal (Custom)

The two designs above place their points by a fixed rule. An optimal mixture places them by search. You tell the software how many components you have, from 2 to 10, whether you want a linear or a quadratic Scheffe model, and how many runs your budget allows. It then works through a large pool of candidate blends across the simplex and, using a D-optimal exchange, keeps the subset of runs that measures your chosen model most precisely.



Optimal (Custom) in the catalogue. You set the model and the run budget, and the search picks the blends.

This is the design to use when a lattice or centroid does not fit your situation: you have an awkward number of components, a tight run budget, or you simply want the smallest sensible design for a known model. If you leave the run count blank the software proposes a default of the number of model terms plus ten, which leaves a few spare runs to estimate error. You can never ask for fewer runs than the model has terms; the design screen tells you the minimum as you change the model.

Recipe and process in one study.
If you need to vary the blend and a process setting such as mixing time or temperature together, that is a related but separate design. See Custom designs and import for the Optimal (Combined) design, which crosses a Scheffe mixture model with your process factors.

Reading the run sheet

However the points are chosen, each mixture run reaches the run sheet as real component amounts. The software multiplies each proportion by the total you set, so if your total is 100 mg the columns read in milligrams and every row still sums to 100. Component columns carry the same header style as any other factor, the design letter followed by the name you gave it, so a column might read `A:Lactose (mg)`. Fill in the response columns exactly as you would for any other design, then move to the Analysis branch to fit the Scheffe model.

Design	Components	Places points by	Best when
Simplex Lattice	2 to 12	An even grid of degree m	You want an evenly spread model-fitting design
Simplex Centroid	3 to 10	Equal-parts blends of every subset	Interaction between components is the question

Design	Components	Places points by	Best when
Optimal (Custom)	2 to 10	A D-optimal search on a run budget	Runs are limited or the component count is awkward

What to read next

- A mixture experiment walks one recipe from an empty screen to a recommended blend.
- Custom designs and import covers the combined mixture and process design.
- Fit summary, model and ANOVA explains how the fitted model is judged.

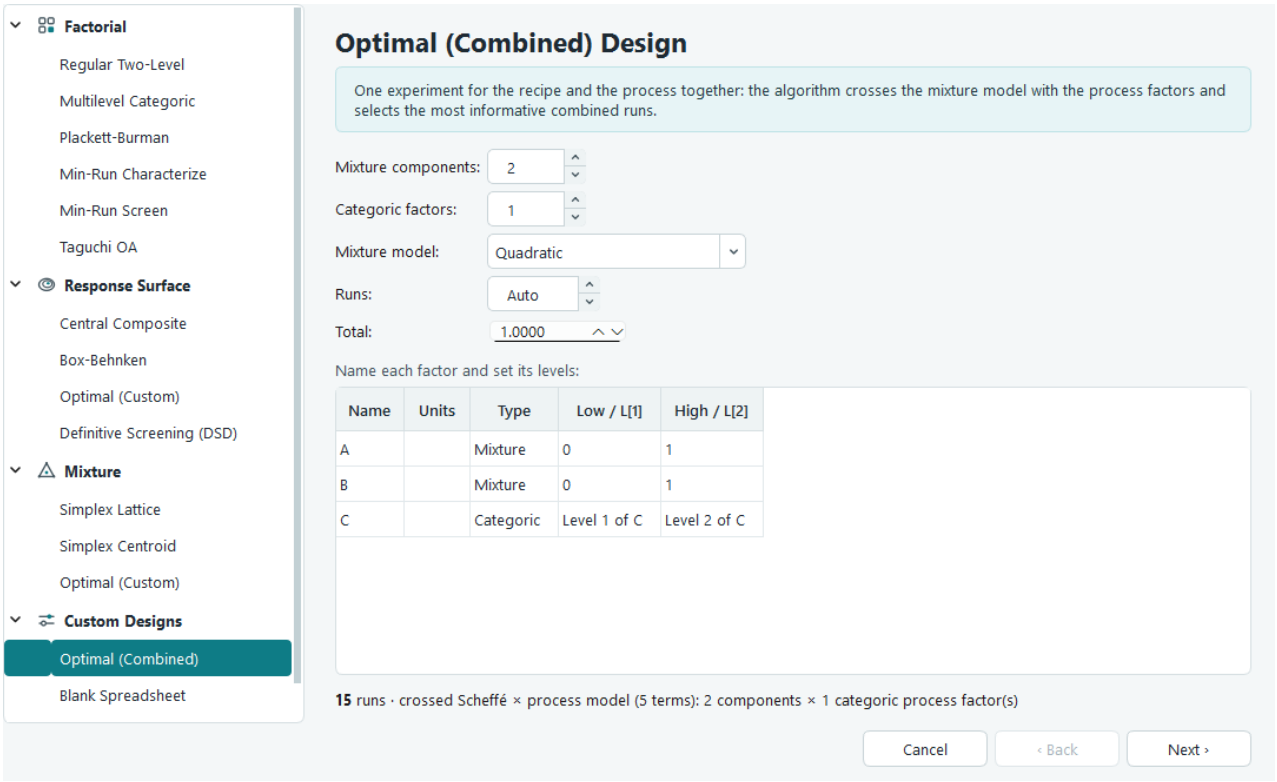
Chapter 9

Custom designs and import

The Custom Designs family is for the cases that do not fit a standard template: one experiment that studies a recipe and a process together, an empty sheet you plan by hand, or a set of results you already have and want to analyse. This page covers all three.

Optimal (Combined)

Sometimes the answer depends on both the recipe and the way you make it. The blend of ingredients matters, and so does a process choice such as the mixer you use or whether a coating step is on or off. Running a separate mixture study and a separate process study wastes runs and hides the interactions between the two. The Optimal (Combined) design puts them in one experiment. It crosses the mixture model with the process factors and asks a Fedorov exchange to pick the most informative combined runs inside the run budget you allow.



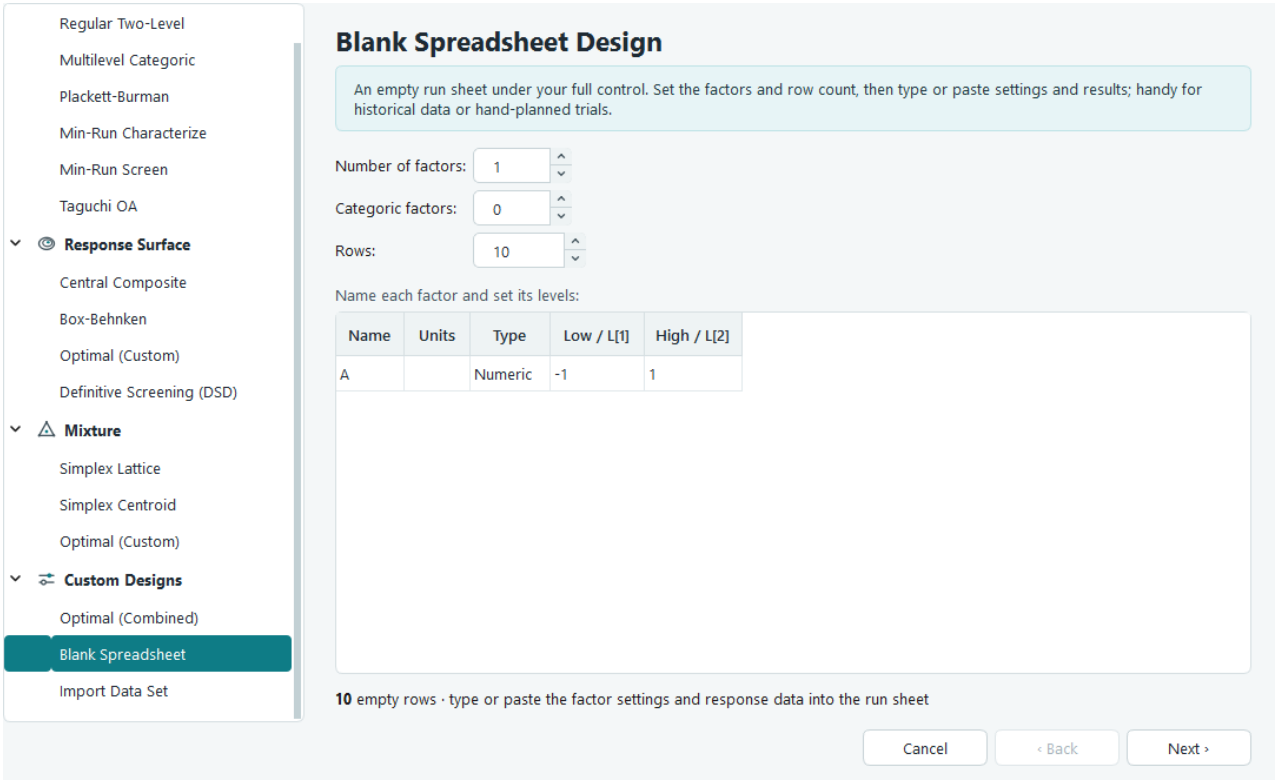
Optimal (Combined). You set the mixture components and at least one process factor, then choose whether to fit a linear or a quadratic mixture model.

On the configuration page you set the number of mixture components (2 to 8) and the number of process factors, which you enter in the categoric factors field. A combined design needs at least one process factor to cross with, so that field starts at one. In the factor table the component rows are marked *Mixture* and run from 0 to the mixture total, and the process rows are marked *Categoric* with two levels each. Pick a mixture model, linear or quadratic, and the software crosses it with the process factors. Leave the run count on *Auto* to get the number of model terms plus ten, or type a fixed number if your budget is set. The line under the table tells you the run count and the term count as you change the options.

How the runs are chosen.
The algorithm builds a pool of candidate blends crossed with every process setting, then keeps the subset that estimates your crossed model with the best precision. Because the run list is search built rather than laid out on a grid, two combined designs with the same settings can differ slightly. That is normal for an optimal design.

Blank Spreadsheet

Not every experiment is planned inside the software. You may already have trials you ran by hand, or historical batches from the plant, or a plan a colleague gave you on paper. The Blank Spreadsheet gives you an empty run sheet with the factors and responses you name, and nothing filled in. You type or paste every setting and every result yourself.



Blank Spreadsheet. Set the number of factors, name them and set their type, choose how many empty rows you want, then fill in the sheet.

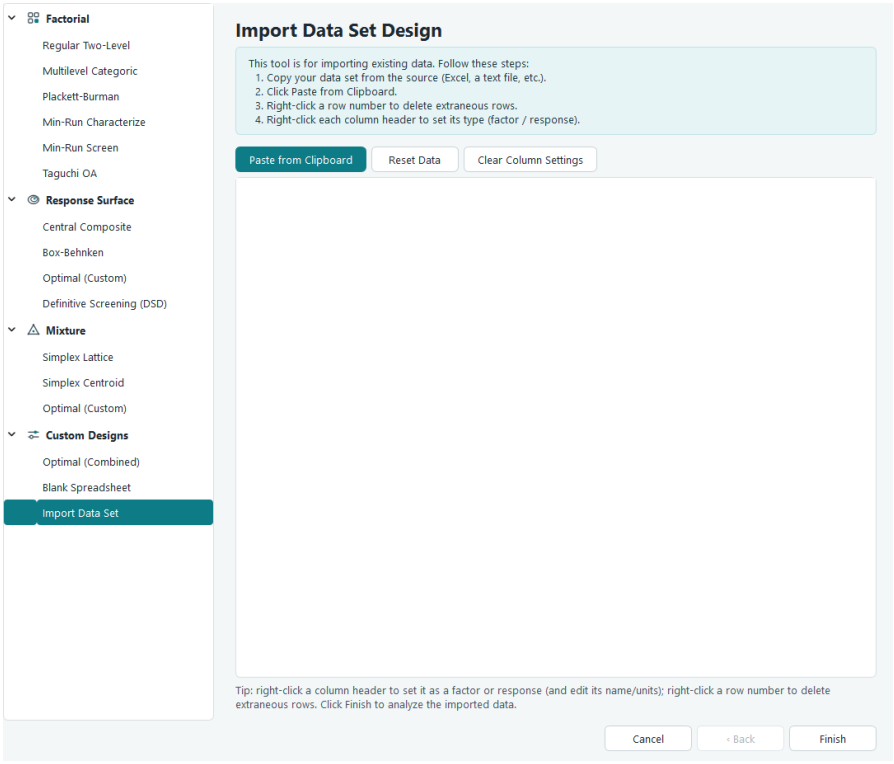
Set the number of factors (1 to 20) and the number of rows you need. Each factor can be numeric, with a low and a high value, or categoric, with two levels; the categoric factors field controls how many of the rows are categoric. Move to the next page to name your responses. When you press Finish you get a run sheet of the size you asked for, all cells empty, ready for you to enter the data. From that point the sheet behaves like any other design: you can paste a whole column from a spreadsheet and undo with `Ctrl` + `Z`.

Blank sheet or import.
Use the Blank Spreadsheet when you want to define the factors and their ranges up front and then key in the numbers. Use Import Data Set below when the numbers already live in a spreadsheet or a text file and you would rather paste the whole block at once.

Import Data Set

Import Data Set reads a table you paste from the clipboard and turns it into a design you can analyse. It is the fastest way to bring in results that were recorded somewhere else. The screen is a single page: paste your data, tidy the rows, tell the software what each column is, then press Finish.

1. Copy the whole block of data from the source, in Excel, a text file, or anywhere that copies tab or comma separated text. A header row of names at the top is fine.
2. Click **Paste from Clipboard**. The software splits the text into rows, and splits each row on tabs if it finds any, otherwise on commas. If the first row looks like text labels sitting above numeric data, it treats that row as a header and drops it, so the labels are not mistaken for a run.



Pasted data. Every column starts unassigned and shows (set type) until you give it a role.

3. Delete anything that is not a run. Right-click a row number and choose **Delete Row**, or select several rows and right-click to delete them together. Blank separator lines, subtotals and stray notes should all go. You can also insert a row if you need to.
4. Set the role of each column. Right-click a column header and open **Set Column Type**, then choose *Numeric Factor*, *Categoric Factor*, *Response* or *Ignore*. To set several columns at once, click their headers with `Ctrl` or `Shift` first, then right-click and apply the type to all of them.
5. Name each column. Right-click a single header and choose **Edit Info** to type its name and units. This is worth doing, because the header names you pasted are not carried over automatically.
6. Click **Finish**. The imported table opens as a normal design workspace, ready to analyse.

Two buttons help you start over without leaving the page. **Reset Data** empties the whole table so you can paste a different block. **Clear Column Settings** keeps the data but returns every column to unassigned, which is useful when you mislabelled the roles and want a clean slate.

What the software works out, and what it cannot

The importer reads structure, not meaning. It is worth knowing exactly where the line falls so you are not surprised by the result.

The software works this out for you	You have to tell it this
Whether the block is tab or comma separated, and where the rows and columns fall.	Which columns are factors and which are responses. Every column starts with no role.
Whether the top line is a header of labels, which it then drops.	The name and units of each column. Defaults are the plain factor letters A, B and so on, and R1, R2 for responses, until you set names with Edit Info.
The range of a numeric factor, taken as the lowest and highest value in that column.	Whether a column of numbers is really a numeric factor or a categoric one written as codes. Set it as Categoric if the numbers are labels.
The levels of a categoric factor, taken as the distinct text values in that column.	The order of the runs. Rows are kept exactly as pasted; there is no randomisation.

Numeric factors need variation.
A numeric factor column must contain at least one number, and its values must not all be equal. If a column is constant the software cannot work out a range and Finish will stop with a message. Cells that are not numbers in a numeric factor or a response are read as missing.

Analysing imported data

An imported data set analyses exactly like a design the software built for you. You can fit a model, read the ANOVA, check the residuals and draw the model graphs, all in the usual Analysis tabs. There is one honest caveat. A built design places its runs on purpose so that the model terms can be told apart cleanly. Your imported runs were chosen for other reasons, so they may not have that property.

This shows up under Information, in the Evaluation screen. For an imported set it may report high variance inflation, which means two terms move together and their effects are hard to separate; it may list terms the design cannot estimate at all; and it may show uneven leverage, where a single run has too much pull on the fit. None of that is a fault in the software. It is telling you the truth about data that was not collected to a plan. Read Evaluating a design to understand each measure, and treat a weak evaluation as a reason to be careful with the conclusions, not as an error to fix.

Part 3

Working with a design

Chapter 10

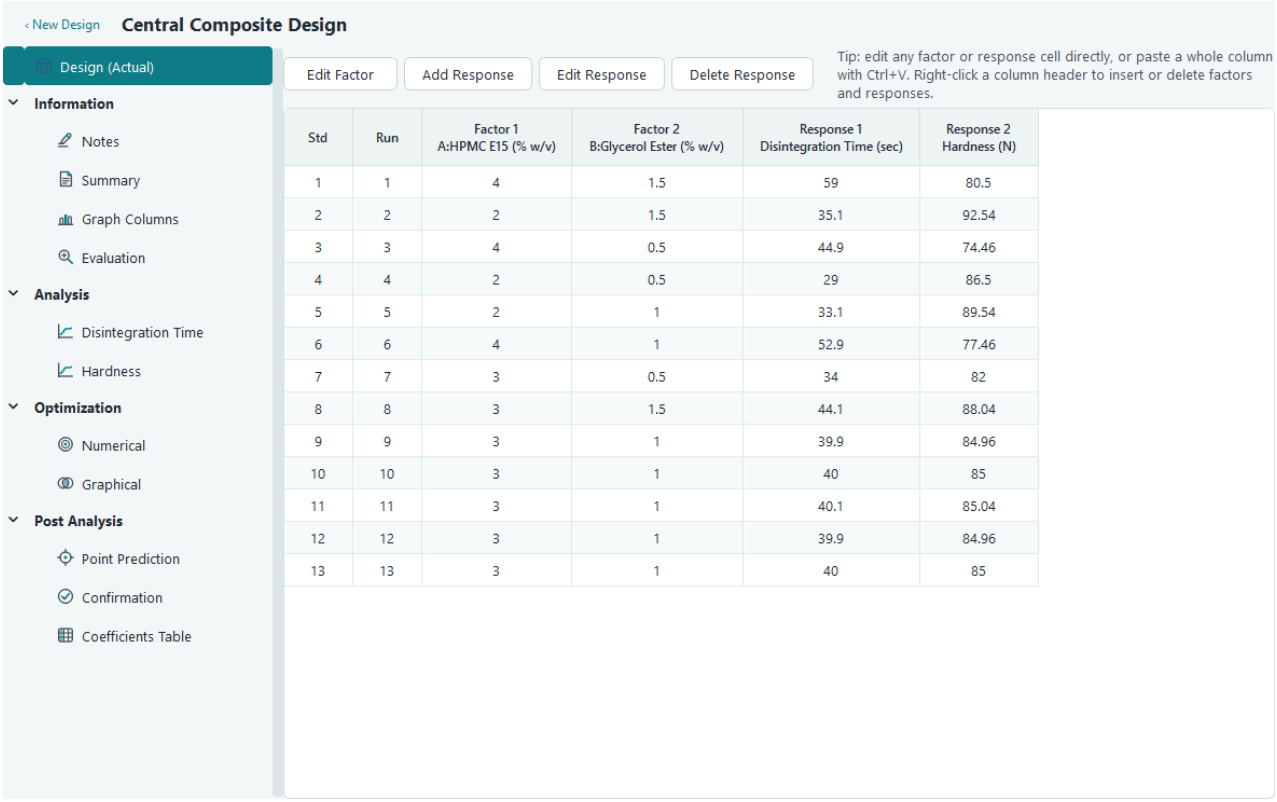
The design workspace

The workspace is where a built design lives. The tree on the left is the table of contents for one experiment; the panel on the right shows whichever page you select. This reference covers the run sheet and the Information branch. Analysis and optimization each have their own pages.

When a design is built, or when you open a saved `.ogdx` file, you land on **Design (Actual)**, the top node of the tree. That node holds the run sheet, the single most important screen in the software. Everything below it reads from the same design, so the moment you change a value here, the summary, the plots and the analyses see the change too.

The run sheet

Each row of the run sheet is one experiment. The software has already worked out the factor settings for every run; your job is to carry out the runs and type the measured responses into the last columns. The worked example used throughout this documentation is a face centred central composite design on two factors, A: HPMC E15 (2 to 4 % w/v) and B: Glycerol Ester (0.5 to 1.5 % w/v), with 13 runs including 5 centre points, and two responses, Disintegration Time (sec) and Hardness (N).



Std	Run	Factor 1 A:HPMC E15 (% w/v)	Factor 2 B:Glycerol Ester (% w/v)	Response 1 Disintegration Time (sec)	Response 2 Hardness (N)
1	1	4	1.5	59	80.5
2	2	2	1.5	35.1	92.54
3	3	4	0.5	44.9	74.46
4	4	2	0.5	29	86.5
5	5	2	1	33.1	89.54
6	6	4	1	52.9	77.46
7	7	3	0.5	34	82
8	8	3	1.5	44.1	88.04
9	9	3	1	39.9	84.96
10	10	3	1	40	85
11	11	3	1	40.1	85.04
12	12	3	1	39.9	84.96
13	13	3	1	40	85

The run sheet. Column headers carry the factor letter and the name you gave it, so `A:HPMC E15 (% w/v)` is factor A.

Std, Run and the point of randomising

The first two columns are fixed and cannot be edited. **Std** is the standard order, the tidy textbook order in which the design was generated. **Run** is the order in which you should actually perform the experiments, and the rows are shown sorted by it.

> **Run order is randomised on purpose.**
> If anything drifts while you work, a warming room, an ageing reagent, an instrument slowly going out of calibration, a fixed run order would let that drift line up with a factor and masquerade as a real effect. Randomising breaks that link, so run the experiments in Run order, not in Std order.

If your design is split into blocks, a **Block** column appears between Run and the first factor. It too is read only. The two factor example has no blocks, so you will not see that column there.

Entering and pasting response data

Click a response cell and start typing; you do not need to double click first. Response cells and numeric factor cells accept numbers only, so a stray letter is simply rejected. To bring in a whole column from a spreadsheet, copy it there, click the top response cell here and press `Ctrl` + `V`. Paste fills downward and across, following the tab and newline layout of what you copied, and it quietly skips the Std, Run and Block columns so it will not overwrite them.

> **Paste lands on a response.**
> If the cursor is not already on a response column when you paste, the data drops into the first response column instead. Add at least one response before pasting, or the software will tell you there is nowhere to put the numbers.

Cut, copy, paste, undo and redo

The run sheet uses the standard Edit menu and toolbar, so the shortcuts are the ones you already know. Copy and cut work on the block of cells you have selected. Delete clears the selected factor and response cells but leaves Std, Run and Block alone. Undo remembers a long history of cell edits, so you can walk mistakes back one step at a time.

Action	Shortcut	What it does here
Copy	`Ctrl` + `C`	Copies the selected cells to the clipboard.
Cut	`Ctrl` + `X`	Copies, then clears the selected cells.
Paste	`Ctrl` + `V`	Fills cells from the clipboard, skipping Std, Run and Block.
Delete	`Del`	Clears the selected factor and response cells.
Undo	`Ctrl` + `Z`	Steps back through your cell edits.
Redo	`Ctrl` + `Y`	Re-applies an edit you undid.

Right click a cell for the same commands plus **Copy with Headings**, which puts the column titles on the first line so a paste into a report or a spreadsheet arrives already labelled.

Editing factors and responses

You will not always get the design exactly right the first time. You might rename a factor, add a response you forgot, or drop one you no longer need. A small toolbar sits above the run sheet with **Edit Factor**, **Add Response**, **Edit Response** and **Delete Response**. For the rest, right click a column, either a cell or its header, and the menu offers Insert and Delete in place.

Task	How	What changes
Rename a factor	Select its column, press Edit Factor	Its display name and units only. The range and levels were set when the design was built and stay fixed.
Insert a factor	Right click a factor column, Insert Factor, Before or After	Adds a new numeric factor and re-letters the columns A, B, C and so on.
Delete a factor	Right click, Delete Factor	Removes the column from every run and re-letters the rest. A design must keep at least one factor.
Add a response	Add Response, or right click for Insert Response	Asks for a name and units, then adds an empty column.
Edit or delete a response	Select the column, Edit Response or Delete Response	Renames it, or removes the column and its data after a confirmation.

Editing data resets the analysis.
A fitted model is only as good as the numbers it was fitted to, so any change to the data quietly marks the affected analyses out of date. Change a response value and that one response drops back to its plain name and needs **Start Analysis** again. Change a factor value, or add, rename or delete any factor or response, and every response is reset, because a change to the design structure touches them all. Nothing is lost from the run sheet; you simply re-run the fit when you are ready.

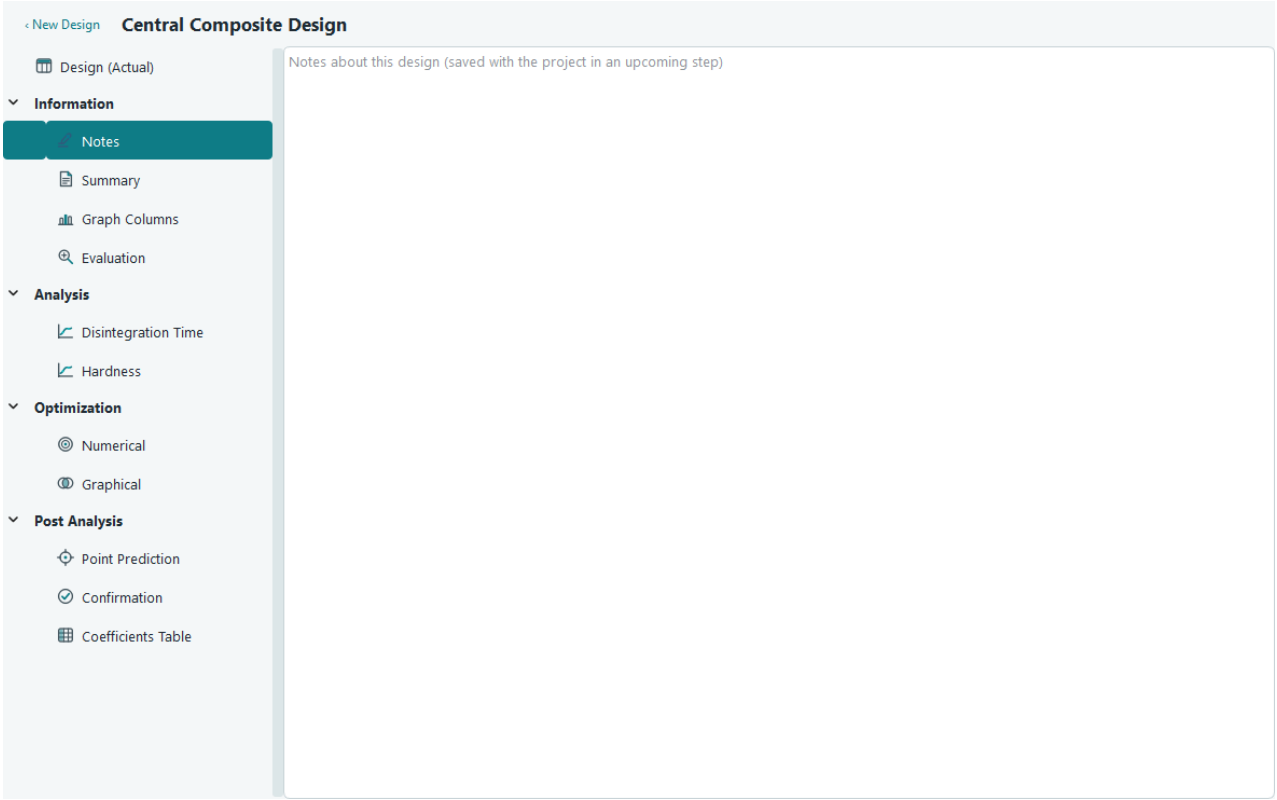
The tree helps you keep track. Under **Analysis** each response is tagged *(Empty)* until it has data, *(Analyzed)* once you have fitted it, and back to its plain name if an edit has outdated the fit.

The Information branch

The Information section gathers everything that describes the design without analysing it: free notes, a computed summary, quick plots of the raw columns, and a formal evaluation. The first three are covered below; the fourth has its own page, Evaluating a design, because it is worth reading before you commit to the runs.

Notes

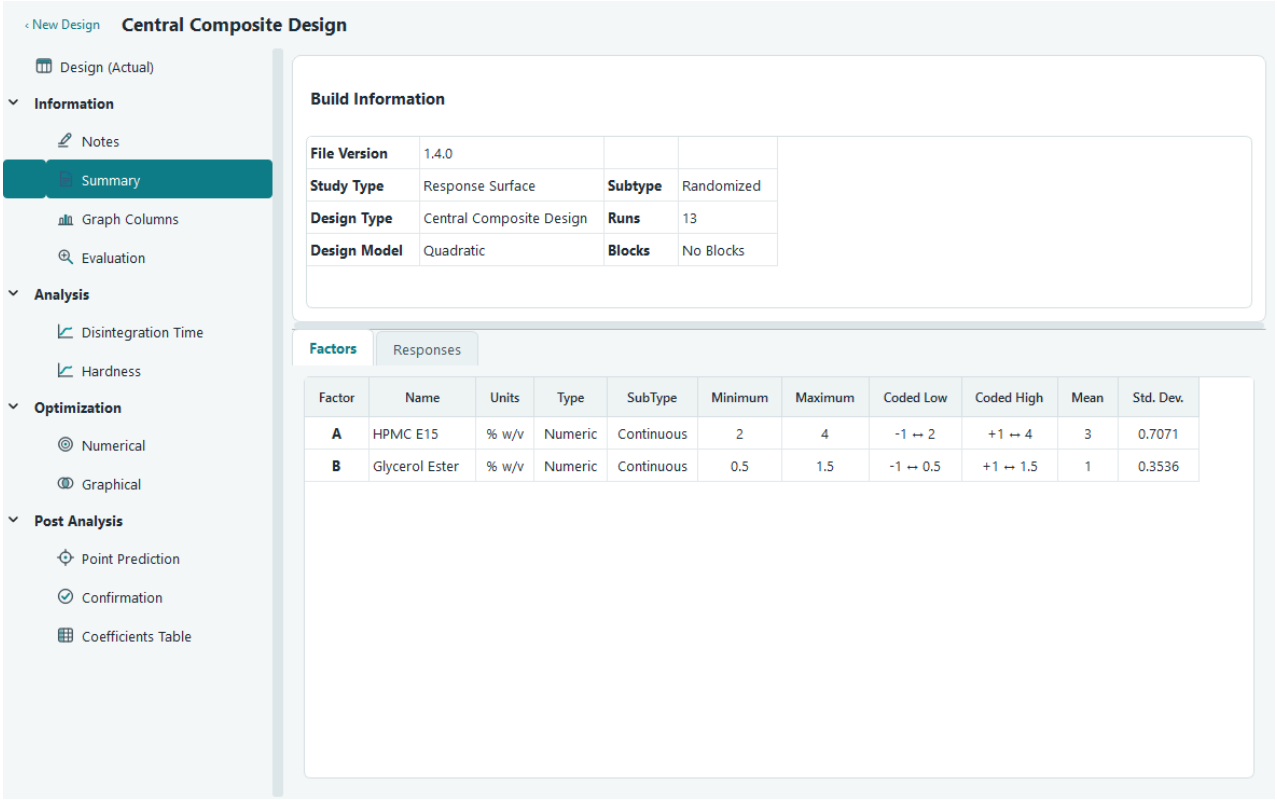
Notes is a plain text area for anything the run sheet cannot hold: what you were trying to learn, which batch of raw material you used, a reminder about a run that went wrong. Write in your own words; there is no structure to follow.



The Notes page. Use it as a lab notebook entry that travels with the design.

Summary

Summary is read only. It reads the design back to you so you can confirm it is what you meant to build. A Build Information panel at the top lists the file version, the study type and its subtype, the design type and run count, the default model, and the block count. Below it, two tabs hold statistics that the software computes straight from the run sheet.



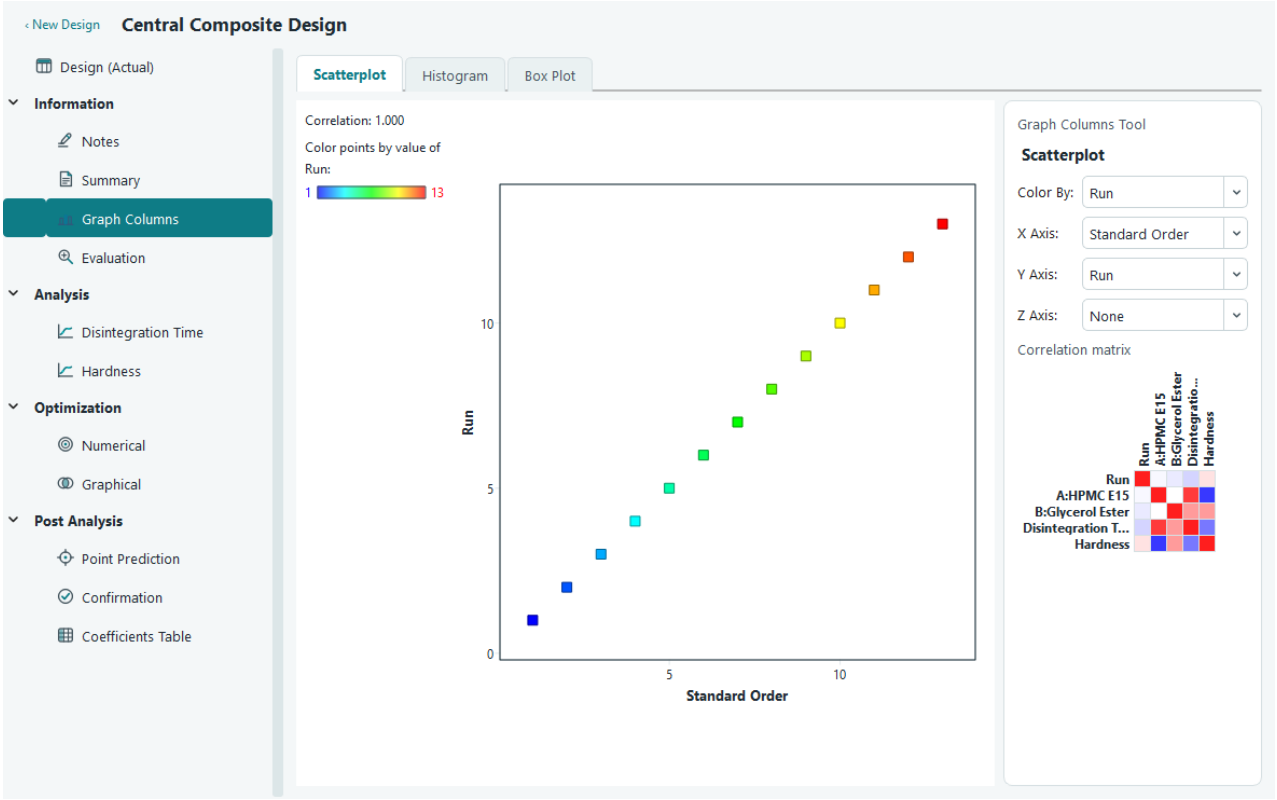
Summary. The Factors tab is shown; the Responses tab sits beside it.

Tab	Columns
Factors	Factor letter, name, units, type and subtype, minimum and maximum, the coded low and high, and the mean and standard deviation of the settings.
Responses	Response label, name, units, the count of observations you have entered, the minimum, maximum, mean and standard deviation, and the max to min ratio.

The response ratio is a quick health check. A large max to min ratio hints that a transform may steady the spread of the data, which is exactly what the Fit Summary tab looks at later in Fit summary, model and ANOVA.

Graph Columns

Graph Columns lets you look at the raw numbers before you fit anything to them. It is the fastest way to catch a typo, a wildly out of range value or a pair of columns that move together. Pick a chart from the tabs across the top, then set the columns from the tool panel on the right. Every column is available: Standard Order, Run, each numeric factor and each response.



Graph Columns. The scatterplot is chosen, with the correlation matrix below the controls.

- **Scatterplot** plots any column against any other and can colour the points by a third. The panel reports the correlation between the two axes so you can put a number on what you see.
- **Histogram** shows the shape of one column as frequency or as relative percent. Set the number of bins, or press Default to let the software choose from the count of values, and tick the box to print the count above each bar.
- **Box Plot** draws the median, the quartiles and the whiskers of one column, with outliers marked as points. Choose a grouping column for the X axis and the box is split by its levels, so you can compare the response across the settings of a factor.

Under the scatterplot controls sits a **correlation matrix**, a coloured grid of the pairwise correlations across Run, the factors and the responses. Red means the pair moves together, blue means it moves in opposite directions, and the deeper the colour the stronger the link. Hover any cell to read the exact value. In a well built design the factor columns should barely correlate with each other; strong colour between two factors is worth a second look.

Plots follow your data.
Graph Columns refreshes whenever you open it, so it always reflects the latest entries in the run sheet. Right click any plot to copy it or save it as an image.

Saving your work

Save with `Ctrl + S`, or File then Save. The first save asks where to put the file and gives it the `.ogdx` extension; after that, Save writes straight back to the same file and Save As makes a copy. One `.ogdx` file holds the whole design: the factors and responses, every value you have typed, and the analysis state. That last part matters, because when you reopen the file the responses you had analysed are refitted for you, so

you pick up where you left off.

The title bar marks a design with unsaved changes, and the software will offer to save for you before you close the file, open another or leave the program. For the other ways to get data out, a CSV of the run sheet, a printed report or an exported graph, see Saving, printing and exporting.

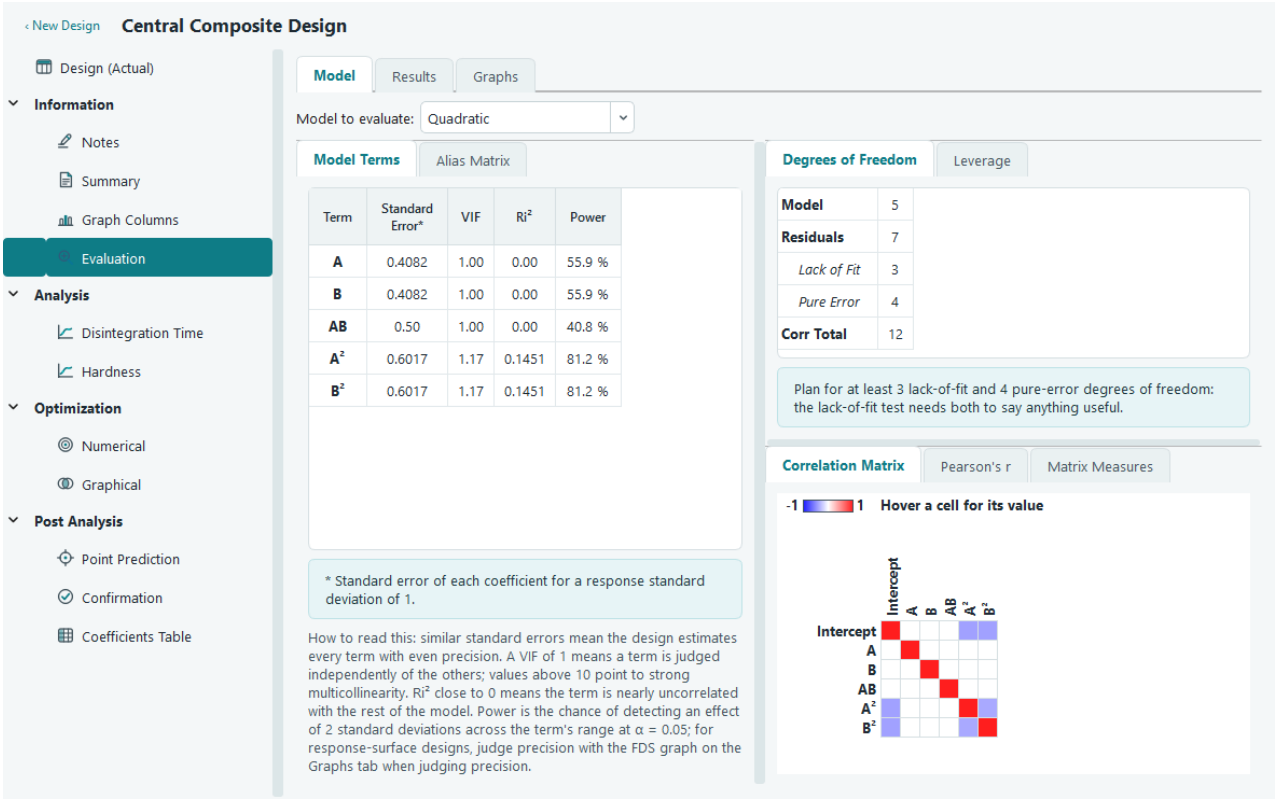
With clean data in the run sheet, you are ready to either check the design first in Evaluating a design, or go straight to fitting a model in Fit summary, model and ANOVA.

Chapter 11

Evaluating a design

Evaluation looks at a design before you run it. It cannot tell you the answer to your experiment, because there are no results yet. What it can tell you is whether the design you are about to run is strong enough to give you a clear answer, while it is still cheap to fix.

You open Evaluation from the **Information** branch of the navigation pane. It studies the pattern of factor settings in the run sheet, nothing else, so it works the moment a design is built and long before any response is measured. The page assumes a model, then asks two questions about it: can this design estimate every term in the model, and how precisely will it do so. Weak spots show up as large numbers in a few key columns, and the fixes are usually simple: add a few runs, add centre points, or fit a smaller model.



The Evaluation page for the worked example, a face-centred central composite design on two factors with 13 runs. The model to evaluate is chosen at the top left.

The page has three tabs across the top: **Model**, **Results** and **Graphs**. The Model tab is the working area with every table. The Results tab is the same information as a plain, printable report. The Graphs tab holds the fraction of design space curve.

Choosing the model to evaluate

Precision is always relative to a model, so the first thing you do is pick one. The drop-down at the top left offers main effects only (linear), two-factor interactions (2FI), and quadratic. The design cannot care what you eventually fit, but the evaluation numbers can change a lot with the model, so evaluate the model you actually intend to analyse. For the response surface example the quadratic model is the right choice, since a central composite design exists precisely to fit curvature.

Degrees of freedom

Degrees of freedom are your experiment's budget. Every run buys you one, and each part of the analysis spends some. The table splits the total into the pieces that matter.

Line	What it counts
Model	Terms the model spends, one per coefficient other than the intercept. More terms need more runs.
Residuals	What is left over to estimate error and to test the model. This is where your safety margin lives.
Lack of Fit	The part of the residual that checks whether the model shape is right. It needs runs at settings that are not centre points.
Pure Error	The part that measures pure noise, from genuine replicates such as repeated centre points.
Corr Total	The whole budget, which is the number of runs minus one.

For the example, the quadratic model spends 5 degrees of freedom on the terms A, B, AB, A squared and B squared. That leaves 7 residual degrees of freedom out of a corrected total of 12. The 5 centre points give 4 pure-error degrees of freedom, and the remaining 3 go to lack of fit.

A useful floor.
Aim for at least 3 lack-of-fit and 4 pure-error degrees of freedom. Below that, the lack-of-fit test has too little to work with and cannot tell you much either way. If either number is small or zero, add runs or add centre points until it rises.

Leverage and average leverage

Leverage measures how much one run pulls the fitted surface toward its own result. A leverage near 1 means the model is forced to pass almost exactly through that point, so a single bad measurement there bends the whole surface and you never see it in the residuals. The table lists the leverage of every run, its space type, and an average at the bottom. The average leverage is simply the number of coefficients divided by the number of runs, so a lower average means the design spreads its effort more evenly. Where you see a leverage close to 1, replicate that run or perform it with extra care.

Per-term precision: standard error, VIF and Ri squared

The Model Terms table has one row per model term and tells you how well each will be estimated. Because there are no results yet, precision is quoted for a response with a standard deviation of 1, so the numbers compare terms and designs rather than predict real units.

Column	How to read it
Standard Error	The uncertainty on that term's coefficient. Similar values down the column mean the design estimates every term with even precision, which is what you want.
VIF	Variance inflation factor. It says how much a term's variance is enlarged because it is correlated with the other terms. A VIF of 1 is ideal: the term is estimated independently. Above 10 signals strong multicollinearity, and that term's coefficient will be hard to trust.

Column	How to read it
Ri squared	How well the other terms predict this one. Close to 0 is good and means the term stands apart from the rest of the model. VIF and Ri squared measure the same problem from two angles.
Power	Your chance of catching a real effect. See below.

A high VIF is usually a sign that two effects move together in your run sheet, so the design cannot pull them apart. Orthogonal designs, such as a clean factorial, show a VIF of 1 on every term. Central composite and other response surface designs carry mild correlation between the squared terms, which is normal and expected.

Power: the chance of detecting an effect

Power answers the practical question, if this factor really does matter, will my experiment notice. The column reports the probability of declaring a term significant at the 0.05 level when its true effect is 2 standard deviations across the factor's range. A power of 90 percent means that nine times out of ten a real effect of that size would be flagged; 40 percent means you would miss it more often than not. Low power comes from too few runs relative to the number of terms, and the cure is more runs.

Power is for factorial-style screening.
Power reads best when you are checking whether a screening design can find the effects you care about. For response surface designs the goal is prediction, not hypothesis tests, so judge those with the fraction of design space graph on the Graphs tab rather than with power alone.

Design quality measures and prediction variance

Below the per-term table, the design quality measures score the design as a whole. The most useful ones describe prediction variance, that is, how uncertain the model's predicted mean is at a point in the factor space, again for a response standard deviation of 1.

Measure	Meaning
Maximum, average and minimum prediction variance	The spread of prediction uncertainty across the region. A small gap between maximum and minimum means the design predicts about equally well everywhere.
G efficiency	How close the worst-case prediction variance comes to the ideal. Higher is better, quoted as a percentage.
Scaled D-optimality	An overall score for how precisely the whole set of coefficients is estimated. Smaller is better.
Determinant and trace of the inverse information matrix	Two more overall precision scores. Smaller values mean tighter coefficients.
Condition number	How near the design is to being unable to separate its terms. Large values warn of numerical trouble and heavy correlation.

Compare like with like.
Scaled D-optimality, the determinant and the trace are only meaningful when you compare two candidate designs that have the same number of runs. On their own the raw numbers mean little.

The fraction of design space curve

Open the Graphs tab for the fraction of design space, or FDS, curve. It is the single best picture of how evenly a response surface design predicts. Read it left to right: the horizontal axis is the fraction of the design space, from none of it to all of it, and the vertical axis is the largest standard error of the predicted mean over that fraction. A point at 0.8 on the curve tells you the error you can expect over the best-covered 80 percent of the region.

You want a curve that is both low and flat. Low means predictions are precise; flat means they are just as precise near the edges as they are in the middle, with no corner of the space left vague. A curve that climbs steeply on the right is warning you that predictions get unreliable toward the extremes, which often means the design needs more runs out at the boundary.

When the numbers look bad

Evaluation is only worth running if you act on it. A few patterns and their usual fixes:

What you see	What to do
VIF well above 10, or a high condition number	Terms are tangled together. Add runs to break the correlation, or fit a smaller model that drops the terms you do not truly need.
Low power on effects you care about	Add runs. Power rises as the residual budget grows.
Zero or tiny lack-of-fit or pure-error degrees of freedom	Add centre points for pure error, and add non-centre runs so lack of fit has something to test.
An FDS curve that is high or rises sharply at the right	Add runs, especially near the edges of the region, or reduce the model order.
A run with leverage near 1	Replicate that run, or perform it with extra care so one measurement cannot bend the fit.

Reducing the model is the fix people forget. A design that struggles with a full quadratic can be perfectly strong for a two-factor-interaction model, so switch the drop-down and look again before you commit to extra runs. Every change you make to the design updates the page the next time you open it, so you can try a fix and see the numbers move before you spend anything in the lab.

Part 4

Analysing results

Chapter 12

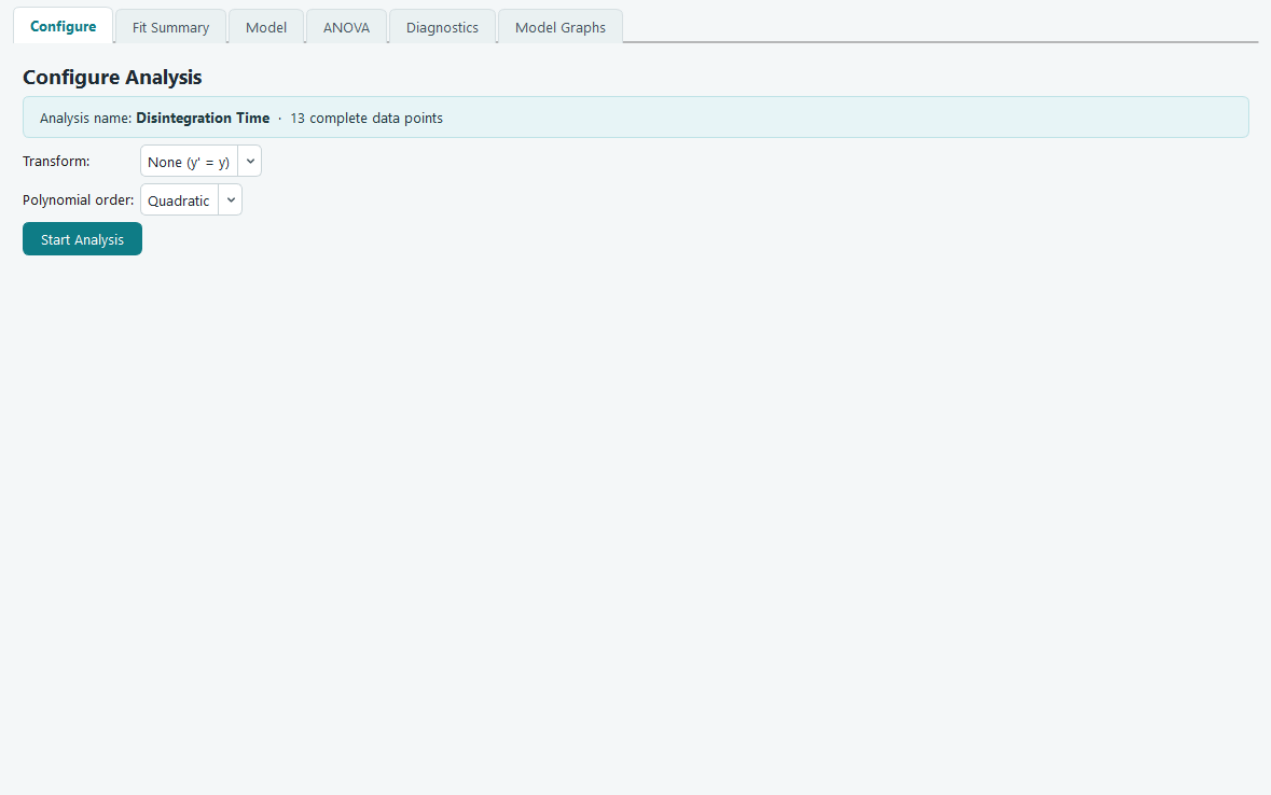
Fit summary, model and ANOVA

This is the heart of the software. You turn a table of measured responses into a fitted model, then judge whether that model is worth trusting. The four tabs covered here run left to right in the order you use them: Configure, Fit Summary, Model, then ANOVA. The examples come from the worked design that ships with the tutorials, a face-centred central composite design on two factors, A: HPMC E15 and B: Glycerol Ester, with Disintegration Time (sec) and Hardness (N) as the two responses.

Open the Analysis branch in the navigation pane and click a response. Each response has its own page, because each one usually needs its own model. The strip of tabs across the top is where you will spend most of your time.

Configure: set up the fit

Before the software can fit anything, it needs to know two things: whether to transform the response, and how curved a model to try. The Configure tab collects both, then hands off to the solver when you press the button.



Configure. The header confirms the analysis name and how many complete data points it found.

Transform decides whether the response is analysed as measured or on a changed scale. The default is `None (y' = y)`, which analyses the numbers exactly as you typed them. Leave it there unless the residuals tell you otherwise. The Diagnostics tab draws a Box-Cox plot that recommends a transform when one would help, so the usual path is to fit once with None, read the diagnostics, and come back only if the plot points elsewhere.

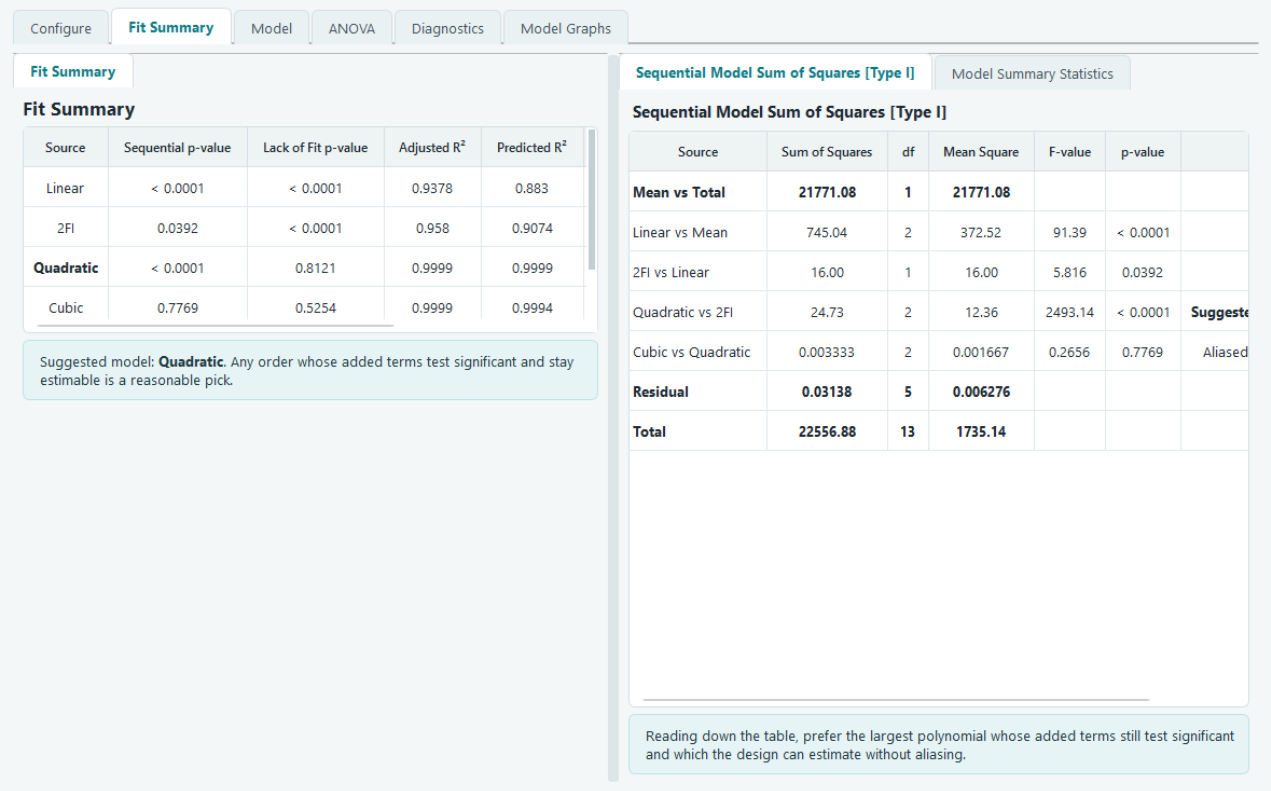
Polynomial order chooses how many kinds of term the model may contain: Linear (main effects only), 2FI (main effects plus two factor interactions), Quadratic (adds squared terms, so the surface can curve), or Cubic. When the page first opens, the software has already looked at your data and preselected the order it suggests, so in most cases you can accept it and move on.

Press **Start Analysis**. The fit runs, and the page moves you to the Fit Summary tab. The button then reports that the analysis is complete and greys out. If you later change the transform or the order, the button wakes up again, because the results on screen no longer match your choices until you re-run.

> **The count matters.**
> Configure reports the number of complete data points, meaning runs that have a value entered for this response. A model needs more complete runs than it has terms, or there is nothing left to estimate the noise. If you see a message about too few runs, either enter more data or pick a smaller model.

Fit Summary: which model does the data support?

You do not have to guess the order. Fit Summary fits Linear, 2FI, Quadratic and Cubic in turn and lays out, side by side, how each one performs. Its job is to steer you toward the simplest model the data can justify.



Fit Summary. The left table ranks the orders; the right panel breaks down the numbers behind them.

The left table has one row per order with four numbers to read:

- **Sequential p-value.** This asks whether the terms that order *adds* to the one below it are worth keeping. A small value (below 0.05) means the extra terms explain real variation, so climbing to this order paid off. A large value means the added terms are noise.
- **Lack of Fit p-value.** This compares the model against the pure error measured from your replicate runs. Here a *large* value is the good one: it says the model does not miss any real pattern. A small value (below 0.05) warns that something systematic is escaping the model.
- **Adjusted R-squared** and **Predicted R-squared.** Both run toward 1. Adjusted R-squared rewards fit but charges for every extra term. Predicted R-squared, built from the PRESS statistic, estimates how well the model forecasts runs it has not seen. You want both high and close to each other.

The right panel shows the working. The **Sequential Model Sum of Squares** table (Type I) reads top to bottom as a staircase: Mean vs Total, then Linear vs Mean, 2FI vs Linear, Quadratic vs 2FI, and so on, each line testing only what its order adds. The **Model Summary Statistics** table lists the standard deviation, the three R-squared measures and PRESS for each order together, so you can compare candidates at a glance.

How the software picks a suggestion

One row is flagged **Suggested**. The choice is not a guess; it is the documented Whitcomb score, the same rule practitioners use by hand. For each order the software forms a score from three ingredients:

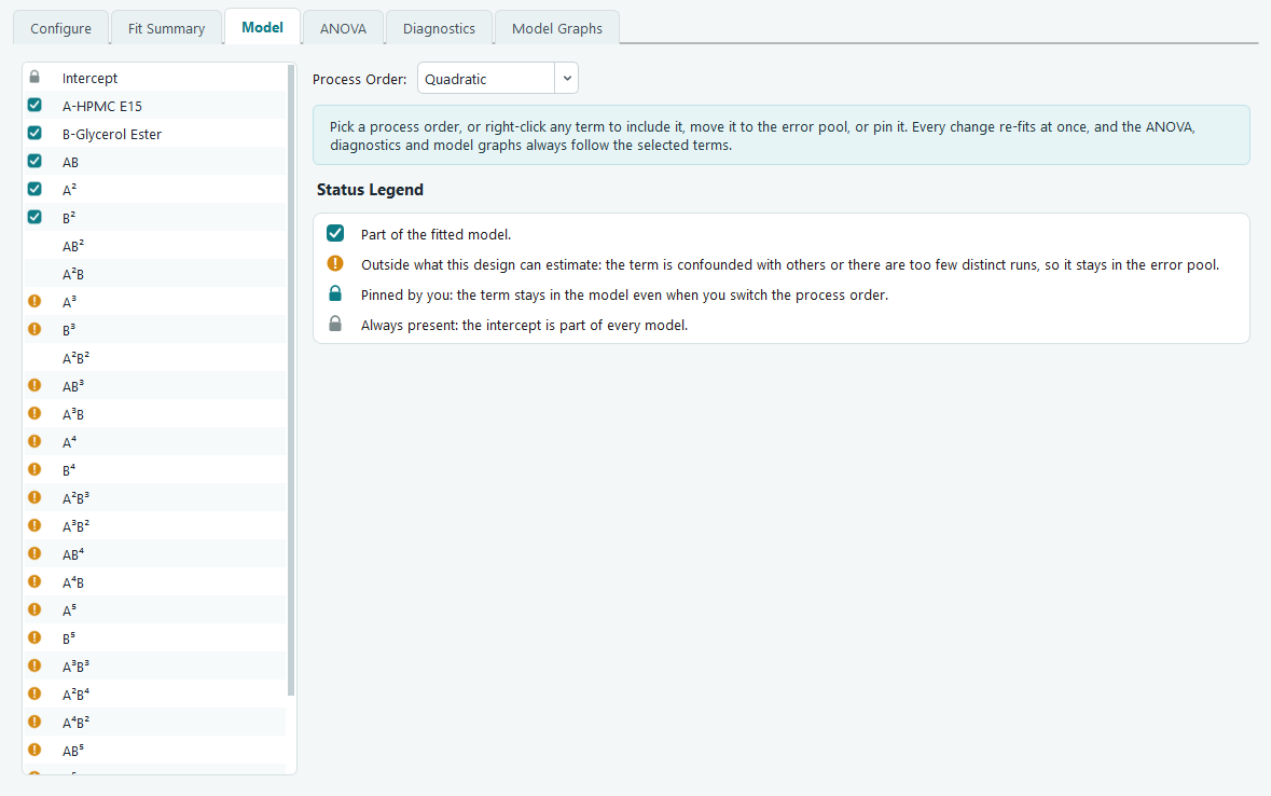
- a sequential-p factor that is 1 when the added terms are significant and shrinks as that p-value climbs above 0.05,
- a lack-of-fit factor that is 1 when lack of fit is comfortably non-significant and shrinks as that p-value drops below 0.10,
- and the model's Predicted or Adjusted R-squared.

Multiply the three together and the order with the best score wins. Because the first factor is set to zero for any order the design cannot fully estimate, an **aliased** model can never be suggested. Aliased rows are labelled as such so you understand why they were skipped. When the Predicted and Adjusted scores disagree, both rows are flagged and the software defaults to the higher order.

> **The suggestion is a starting point.**
> Any order whose added terms test significant and stay estimable is a defensible pick. The suggestion saves you the first decision; the tabs that follow let you refine it.

Model: choose the exact terms

An order like Quadratic is really a bundle of terms. The Model tab opens that bundle so you can add or drop individual terms and watch the fit respond. This is where you trim a term that is not earning its place, or force one in that theory says belongs.



Model. Each term carries an icon that tells you its state at a glance.

The **Process Order** picker on the right is the quick way to set the whole term set: Design Model (the model this design was built to support), Mean, Linear, 2FI, Quadratic, and higher orders where the design has room for them. Choosing one rewrites the term list. The moment you touch an individual term, the picker switches to **Modified** to show the set is now hand-built.

The icon beside each term is its status:

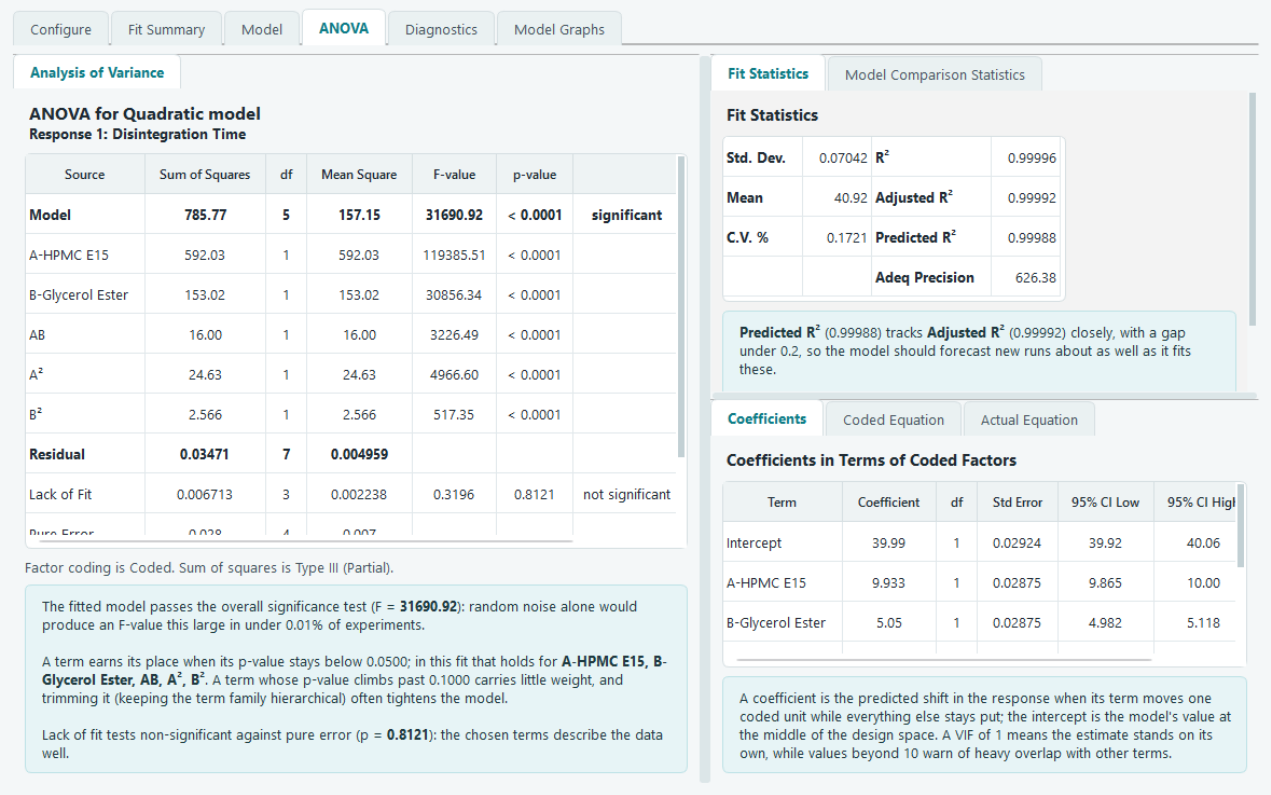
Icon	Meaning
Teal check	The term is part of the fitted model.
Amber exclamation	The design cannot estimate this term: it is confounded with others, or there are too few distinct runs. It stays in the error pool.
Teal padlock	You pinned it. It stays in the model even when you change the process order.
Grey padlock	The intercept, which is part of every model.

Right-click any term (or a selection of them) for **Include in Model**, **Move to Error**, or **Pin Term**. Every change re-fits at once, and the ANOVA, diagnostics and model graphs always follow the terms you have selected here.

> **Hierarchy is kept for you.**
> When you include a term, its lower-order parents come with it, so an interaction never appears without its main effects. When you remove a term, the higher-order terms built on it leave too. Hierarchical models are the ones whose equations behave sensibly, which is why the software enforces this rather than leaving you to police it.

ANOVA: read the verdict

With a model chosen, the ANOVA tab is where you decide whether to believe it. The analysis of variance splits the total variation in the response into the part the model explains and the part left over, then tests each piece.



ANOVA. The sums of squares are Type III (partial), so each term is judged with every other term already in the model.

The main table lists a Model row, then one row per term, then Residual, Lack of Fit, Pure Error and the corrected total. Read it column by column:

- **Sum of Squares** and **df** are the variation attributed to each source and its degrees of freedom. Each model term uses one degree of freedom.
- **Mean Square** is the sum of squares divided by its df.
- **F-value** is the term's mean square over the residual mean square: how large the effect is compared with the noise.
- **p-value** is the chance of seeing an F that large if the term truly did nothing. The Model row is significant when its p-value is below 0.05. A term earns its place when its p-value stays below 0.05; a term whose p-value climbs past 0.10 carries little weight, and dropping it (keeping the family hierarchical) often tightens the model.

The **Lack of Fit** row is the one to check last. It compares the model's misses against the pure error from your replicate centre points. You want it *not* significant: a p-value above 0.05 means the chosen terms describe the data well. If it comes back significant, some real structure sits outside the model, and a higher order or a transform is worth trying. Beneath the table, the software restates the verdict in words, computed from this exact fit, so you are not left translating the numbers yourself.

Fit statistics: what good looks like

The Fit Statistics panel gathers the summary numbers for the chosen model. Here is how to read each one.

Statistic	What it is	What good looks like
Std. Dev.	Root of the residual mean square, in the response's own units.	Small next to the mean.
Mean	Average of the response over the runs.	Context for the others.
C.V. %	The standard deviation as a percent of the mean.	Lower is better; the noise is a small slice of the signal.
R-squared	Fraction of variation the model explains.	Close to 1.
Adjusted R-squared	R-squared with a penalty for every term.	Close to 1 and close to R-squared.
Predicted R-squared	From PRESS: how well it predicts left-out runs.	Close to 1, and within about 0.2 of Adjusted R-squared.
Adeq Precision	A signal-to-noise ratio for the fitted surface.	Above 4.

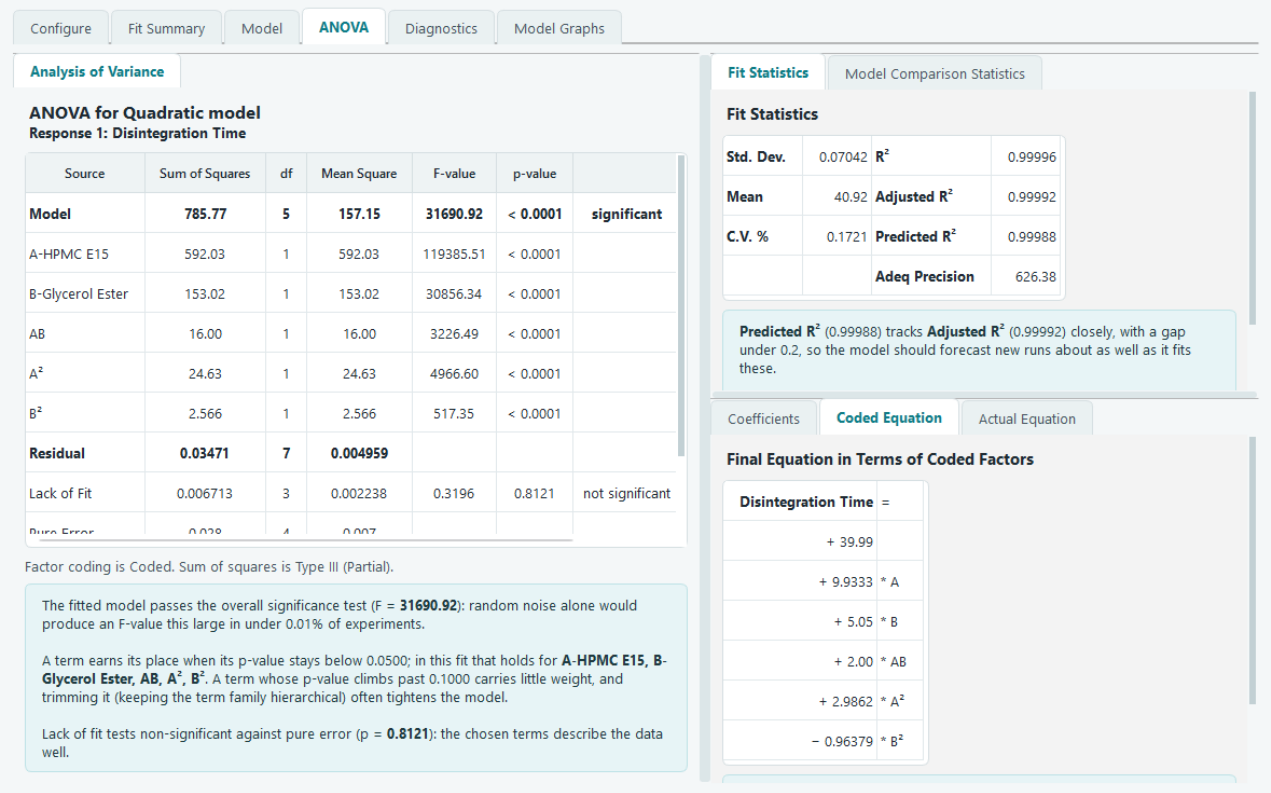
A gap of more than about 0.2 between Adjusted and Predicted R-squared is a hint to look for an outlier, a block effect, or too many terms. The Model Comparison Statistics tab adds PRESS, the log-likelihood, BIC and AICc for weighing one candidate model against another; for those scores, smaller values win.

Coefficients, confidence intervals and VIF

The Coefficients table gives the fitted value of every term along with a 95% confidence interval and a variance inflation factor. A coefficient is the predicted shift in the response when its term moves one coded unit while everything else holds still; the intercept is the model's value at the centre of the design space. If a coefficient's confidence interval straddles zero, that term is not clearly different from zero. The **VIF** measures how much a term overlaps the others: a VIF of 1 means the estimate stands on its own, while values past 10 warn of heavy overlap. For an orthogonal design like this central composite, the interaction and main-effect VIFs sit at or near 1.

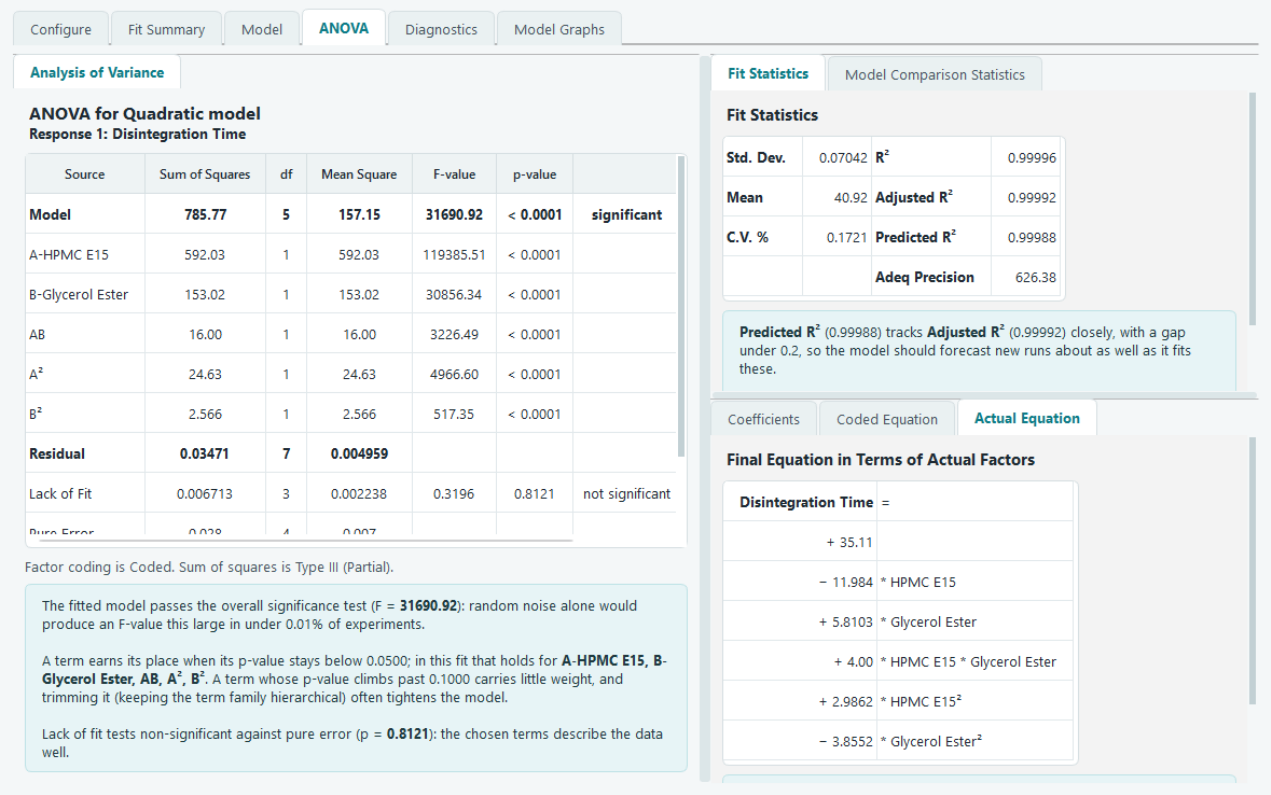
The two equations, and when to use each

Once the model is fit, the software writes it out twice. Both describe the same surface; they differ only in the units their factors take.



The coded equation. Every factor runs from -1 at its low level to +1 at its high level.

Use the **coded equation** to compare importance. Because every factor is on the same -1 to +1 scale, the size of each coefficient tells you directly which factors carry the most weight. This is the form to read when you want to know what matters.



The actual equation. Factors take their real measured values, so you can plug settings straight in.

Use the **actual equation** to predict. It takes factor values in their original units, for example HPMC E15 in % w/v, so you can enter real settings and read off a predicted response. Do not compare its coefficient sizes: the unit scaling inflates or shrinks them, so a big coefficient there can simply mean a factor measured in small numbers. Judge importance from the coded equation, and predict from the actual one.

Both equations follow your terms.
Include or exclude a term on the Model tab and the two equations rewrite themselves along with the ANOVA. What you see here is always the model currently selected, never a stale copy.

With a model you trust, move on to check its residuals before you rely on it.

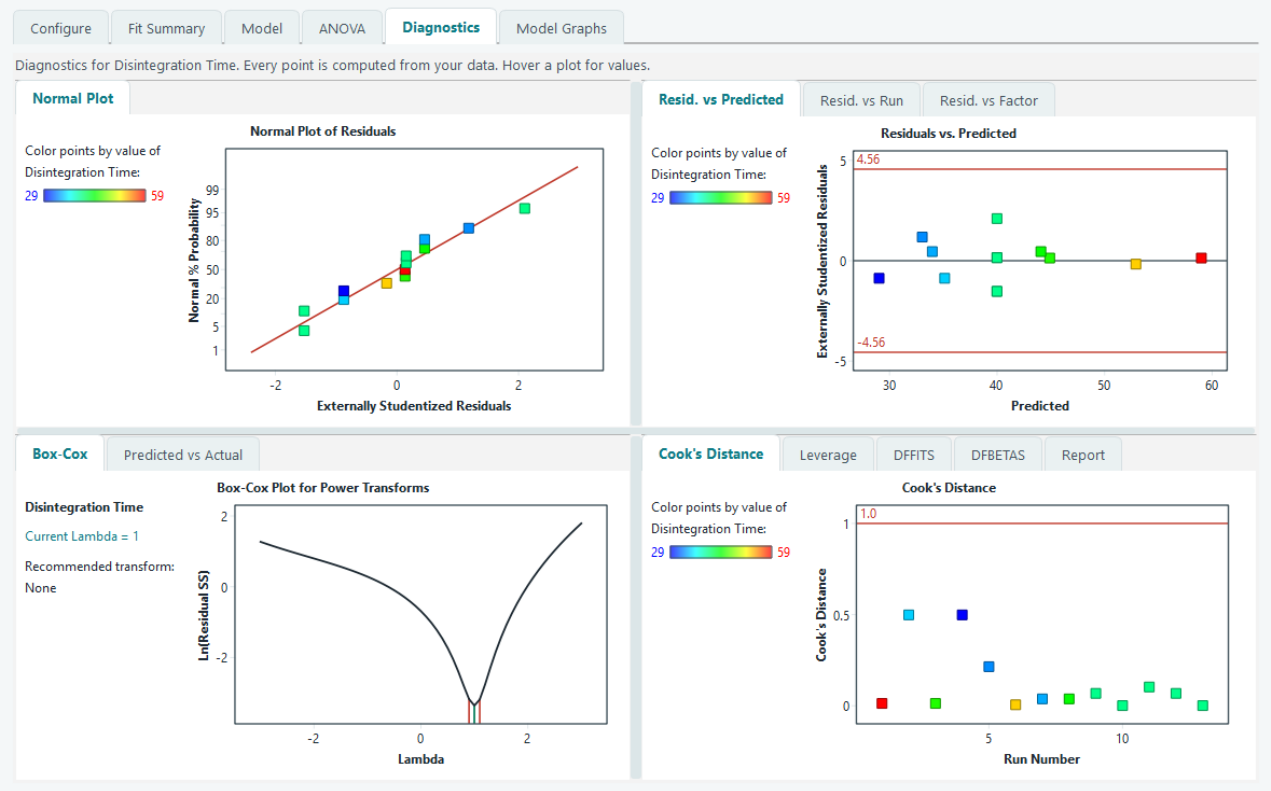
Chapter 13

Diagnostics

The ANOVA tells you the model fits. Diagnostics tell you whether you are allowed to believe it. This tab checks the assumptions behind every p-value, so you catch a bad run or a wrong model before it reaches a report.

Diagnostics is the fifth tab on an analysis page, after Configure, Fit Summary, Model and ANOVA. It exists because least squares makes promises: the residuals are random, they have the same spread everywhere, and they follow a normal distribution. When those promises hold, the ANOVA is trustworthy. When they do not, a small p-value can be an illusion. Every plot here is one way of asking whether the promises hold for your data.

The tab is a grid of four resizable plot panes plus a Report table. Each pane has its own sub-tabs, so a lot of graphs live in a small space. Every point is coloured by its response value, from blue at the low end to red at the high end, which helps you see whether trouble is concentrated in the big or the small results. Hover any point to read its exact numbers. All of this is recomputed from your data the moment you press Start Analysis on the Configure tab.



The Diagnostics tab for Disintegration Time in the worked HPMC E15 and Glycerol Ester example. Four plot panes, each with sub-tabs, and a Report tab.

One word first: studentized.
Most of these plots use the *externally studentized residual*, not the raw residual. That is the residual for a run expressed in standard deviations, using an error estimate that leaves that same run out of the calculation, so a genuine outlier cannot quietly shrink the yardstick it is being measured against.

Normal plot of residuals

This is the first assumption to check. The plot puts the externally studentized residuals on the horizontal axis and their normal probability on the vertical axis, with the vertical scale stretched so that a normal distribution comes out as a straight line. A reference line is drawn for you to compare against.

- **Points close to the straight line.** The residuals are near normal. This is what you want, and minor wobble at the ends is normal and not a concern.
- **An S shape, or a clear bend.** The residuals are skewed rather than symmetric. That usually means the response should be transformed. Read the Box-Cox plot below before you do anything.
- **One point stranded far out to the left or right.** A possible outlier. Note its colour and go find it in the Report table.

Residuals versus predicted, run and factor

The second pane holds three related plots, each with the externally studentized residual on the vertical axis and a zero line through the middle. Red control lines mark the outlier limit: any point beyond them is worth a look. The limit is a Bonferroni corrected t value, computed from your design rather than fixed in advance. It takes account of the number of runs and the residual degrees of freedom, so the cutoff fits the size of your experiment instead of being one blanket number.

- **Residuals versus Predicted.** You want a shapeless, even band of points around zero. A funnel or megaphone, where the spread grows as the predicted value grows, means the variance is not constant. That breaks an assumption and is the classic sign that a transform is needed.
- **Residuals versus Run.** The points are joined in run order. A trend, a drift, or a step partway through means something changed over the course of the experiment, a warming room or an ageing reagent for example. This is exactly why the run order is randomised, so such drift shows up here instead of masquerading as a factor effect.
- **Residuals versus Factor.** Pick a factor from the box. A curved pattern here means that factor has curvature the model did not capture, which is a hint to raise the polynomial order on Configure.

Box-Cox plot for a transform

When the plots above suggest a transform, the Box-Cox plot tells you which one. It refits the model at many powers of the response and plots the natural log of the residual sum of squares against the power, called lambda. The lowest point on the curve is the transform the data likes best.

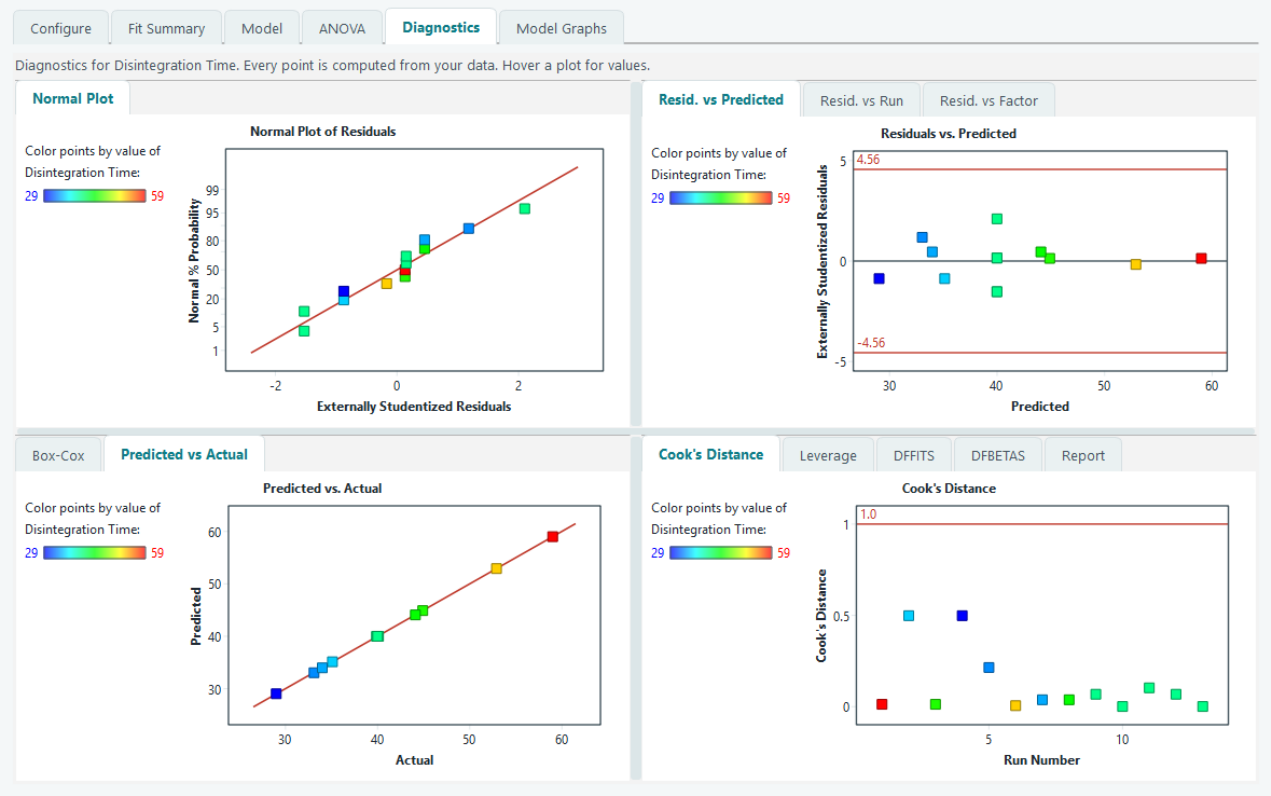
Three markers make it readable. A blue line sits at your **current lambda**, which is the transform set on the Configure tab. Out of the box the response is untransformed, so the current lambda is 1. A green line sits at the **best lambda**, the value at the bottom of the curve. Two red lines mark a 95 percent confidence interval around the best value. The text panel on the left states the current lambda and a plain recommendation, such as Natural Log or Square Root or None.

How to read it: if the blue current line falls inside the red interval, no transform is needed and the recommendation reads None. If the blue line sits outside the interval, the response would benefit from the transform named in the panel. Box-Cox needs every response value to be positive, so if a response contains a zero or a negative number the plot explains that instead of drawing a curve.

> **Read the recommendation as advice, not an order.**
> A transform can straighten the normal plot and flatten a megaphone in one move, but it also changes how the model reads. Only reach for one when a diagnostic actually points to it, not out of habit.

Predicted versus actual

This plot, shared with the Box-Cox pane, is the simplest honesty check in the tab. It puts your measured values on the horizontal axis and the model's predictions on the vertical axis, with a diagonal reference line for a perfect fit. Points hugging the diagonal mean the model predicts what you actually saw. A point that sits well off the line is one the model struggles with, and its colour tells you at once whether the misses are among the high or the low results.



Predicted versus actual for the worked example. Tight clustering along the diagonal is a good sign.

Influence: who is steering the model

A model is fitted to all the runs at once, but not every run pulls with equal force. The fourth pane holds four influence measures, each plotted against run number with a threshold line that the software draws for you. A point past its line is not automatically wrong; it is simply a run with enough sway that you should know it is there.

Measure	What it asks	Line the software draws
Cook's Distance	How much every prediction would shift if this one run were removed.	A flag at 1.0.
Leverage	How unusual this run's factor settings are, on their own, before the response is even looked at.	Twice the average leverage, 2p divided by n, capped at 1, where p is the number of model terms and n the number of runs.
DFFITS	How far this run's own prediction moves when the run is left out, in standard errors.	Plus and minus 2 times the square root of p divided by n.

Measure	What it asks	Line the software draws
DFBETAS	How much one chosen coefficient changes when this run is left out. Pick the term from the box.	Plus and minus 2 divided by the square root of n.

Leverage deserves a word. In the worked face-centred central composite design, the runs sitting on the edges of the factor ranges naturally carry more leverage than the five centre points, because their settings are further from the middle. High leverage on its own is a feature of a good design, not a fault. It only becomes a worry when a high leverage run also has a large residual, because then it is both influential and poorly fitted.

What to do when a point is flagged

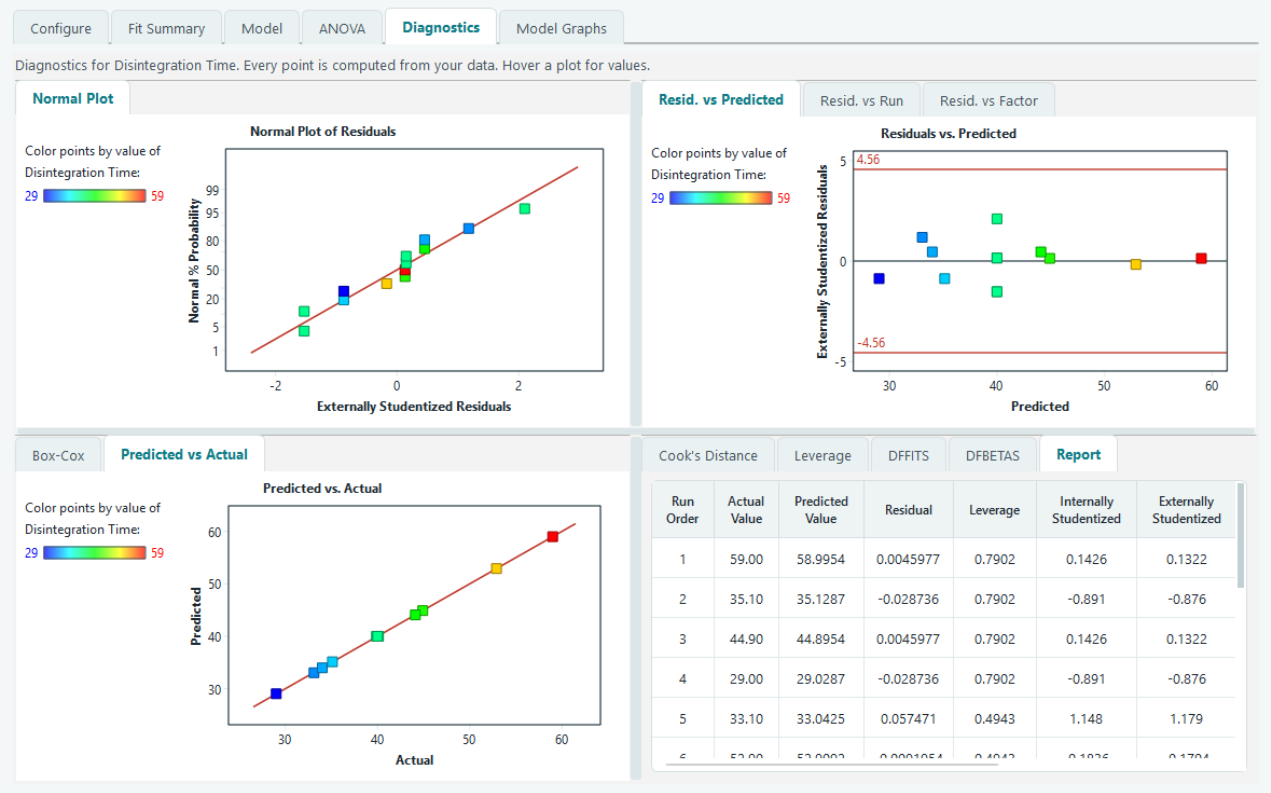
A flag is a question, not a verdict. Work through it in this order.

1. **Check the data first.** Open the run sheet and look at that run. A transposed digit, a wrong unit, a value typed into the wrong response: these account for most flagged points and are fixed in seconds.
2. **Ask what happened at the bench.** If the number is typed correctly, think back to that run. A spill, a mislabelled sample or an instrument that misbehaved is a real reason to treat the run with suspicion.
3. **Repeat the run if you can.** The cleanest resolution to a doubtful point is to run it again and see which value is real.
4. **Only then consider setting it aside,** and say so plainly in your Notes. Removing a run changes every result on every tab.

> **Do not delete a run because it is inconvenient.**
> A point that disagrees with your expectation is often the most informative one in the whole design. Remove data only when you have a specific, documented reason the run is invalid, never merely to make a p-value cross a threshold.

The Report table

The Report sub-tab lays out every per run diagnostic as one row per run, so you can scan the whole design at once and match a flagged plot point to its actual numbers. The columns are Run Order, Actual Value, Predicted Value, Residual, Leverage, Internally Studentized, Externally Studentized, Cook's Distance, DFFITS and Std Order. Two columns highlight problems in red for you: the externally studentized residual when it passes the outlier limit, and DFFITS when it passes its threshold. Those red cells are your shortlist of runs to investigate.



The Report table. Red values mark runs that crossed a threshold and are worth a second look.

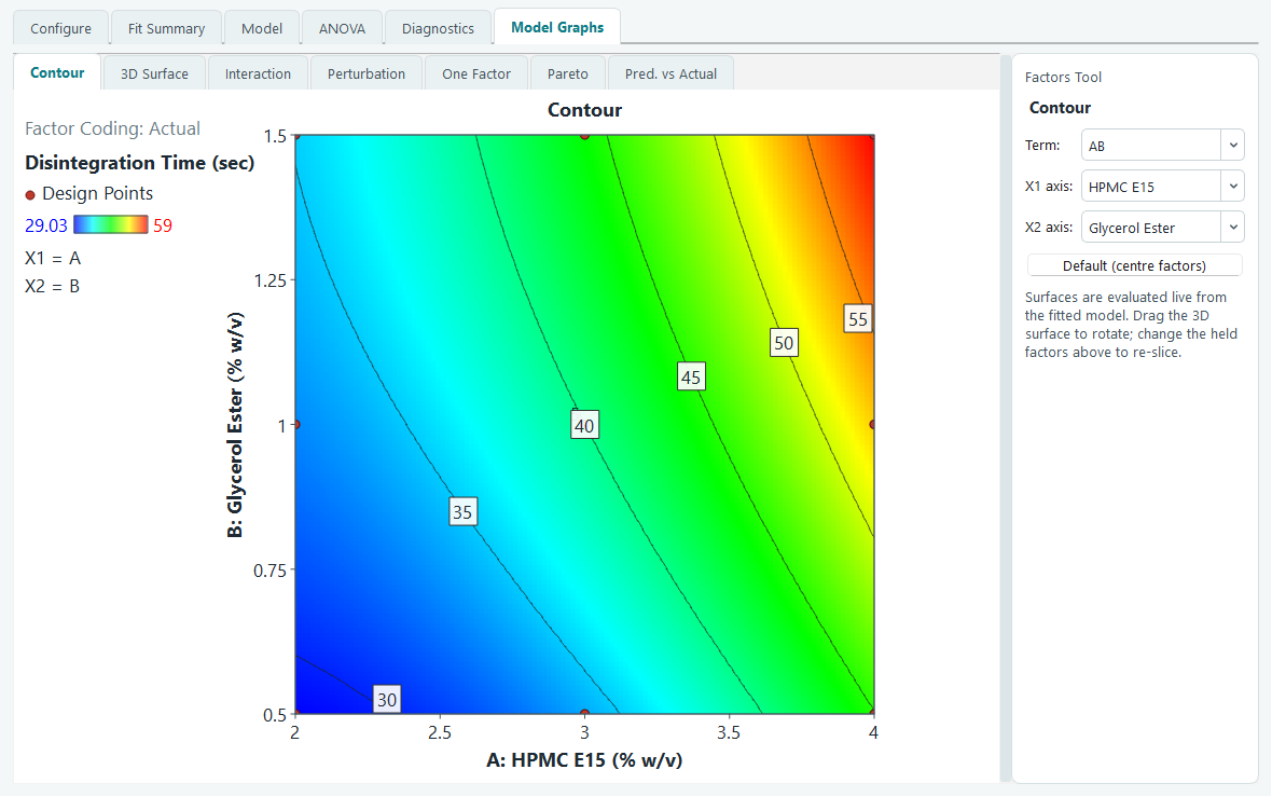
When the plots are clean, the normal plot is straight, the residuals are an even band, no point is flagged and Box-Cox recommends None, you can trust the ANOVA and move on. The next tab draws the model you have just validated.

Chapter 14

Model graphs

The Model Graphs tab draws the fitted model as a picture. The ANOVA tells you a term matters; a model graph shows you what it does. This page explains every graph on the tab, how to read it, and how the Factors Tool on the right decides what you are looking at.

Model Graphs is the last tab in the analysis strip, after Diagnostics. It only has anything to show once you have run Start Analysis, because every curve and surface is evaluated live from the model you fitted. The tab is split into two parts: a row of graph tabs on the left, and the **Factors Tool** on the right that controls all of them.



The Model Graphs tab. The graph tabs run Contour, 3D Surface, Interaction, Perturbation, One Factor and Predicted versus Actual. The Factors Tool on the right sets what each one displays.

The example used throughout this page is the worked one from the tutorials: a face-centred central composite design with two numeric factors, A: HPMC E15 (2 to 4 % w/v) and B: Glycerol Ester (0.5 to 1.5 % w/v), 13 runs including 5 centre points, analysed for Disintegration Time in seconds.

The Factors Tool: axes and slices

A response surface lives in as many dimensions as you have factors, but a graph is flat. The Factors Tool is how you take a readable two-dimensional slice through it. It changes to match the graph you are on, so it never shows a control that does not apply.

- **Which factors go on the axes.** For the Contour, 3D Surface and Interaction graphs you pick two: X1 and X2. The *Term* box is a shortcut that sets both at once from a list of factor pairs (AB and so on). For the One Factor graph you pick a single factor.

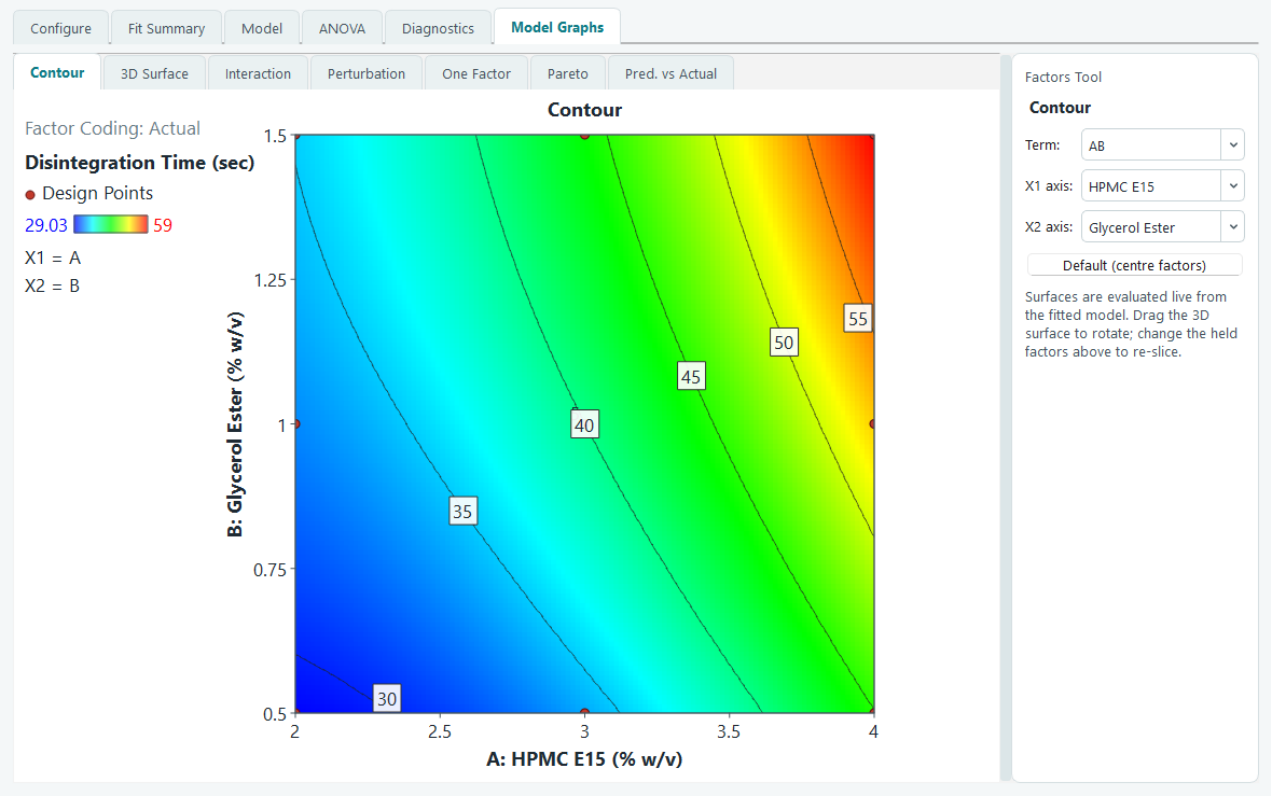
- **Where the other factors sit.** Every factor that is not on an axis gets a slider labelled with its letter and name. The slider runs in coded units from -1 (the low setting) to +1 (the high setting), and the number beside it shows the actual value you are holding it at. Drag a slider and the graph re-slices immediately, so you can watch the surface change as you move a held factor.

The *Default (centre factors)* button returns every held factor to its centre, which is the natural place to start. Because the example has only two factors and both sit on the axes, it has no held factors and no sliders; the tool becomes busy once you have three or more.

> **Design points follow the slice.**
> A run is only drawn on a graph when it genuinely lies in the plane you are showing, meaning every factor that is not on an axis sits exactly at the value the sliders are holding. Move a held factor off a level the design actually used and its points disappear, because none of your runs were performed there. This is deliberate: it stops the software from pretending a point sits on a slice it was never measured on. With the two-factor example every run lies on the plane, so all 13 are always visible.

Contour

The contour map is the one people put in reports. It looks straight down on the response surface. X1 runs left to right, X2 runs bottom to top, both in their real units, and the response is shown by colour: cool colours for low values, warm colours for high. Curved iso-lines join points of equal response, each labelled with its value, so you can read a target level straight off the map.



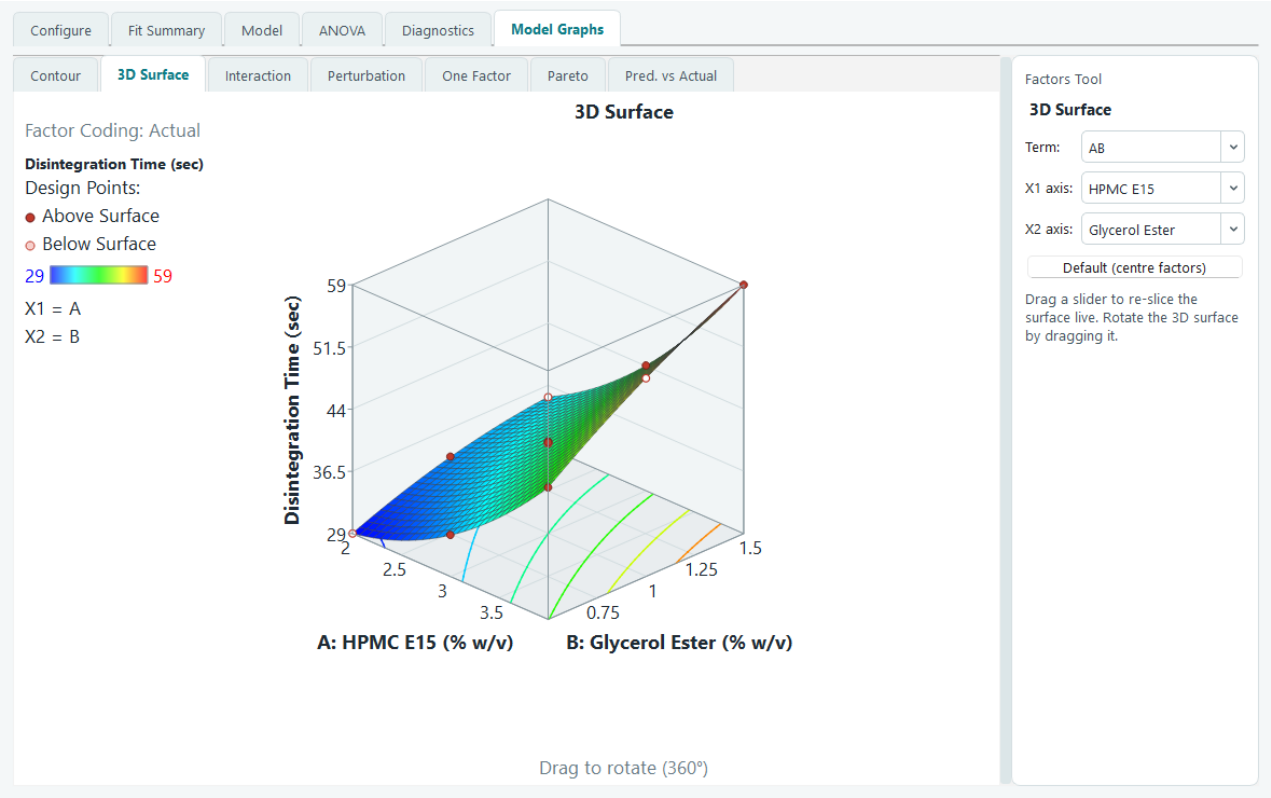
Contour map for Disintegration Time. Red dots are the design points; where several runs share a location a small count is shown for the replicates.

The red dots are your runs, placed at their factor settings. Move the mouse over the map and a small read-out shows the predicted response at the exact spot under the cursor, which is handy for reading a value between the labelled lines. The panel on the left records the response, the colour scale from low to high, and the actual

values of any held factors.

3D Surface

The 3D surface shows the same information as the contour map, but lifted into a solid shape, which is the view that makes a peak, a valley or a saddle obvious at a glance. The surface is colour-shaded by height, the floor beneath it carries the same contour lines projected down, and your runs are drawn as dots: filled red where a run sits above the fitted surface, hollow where it sits below.



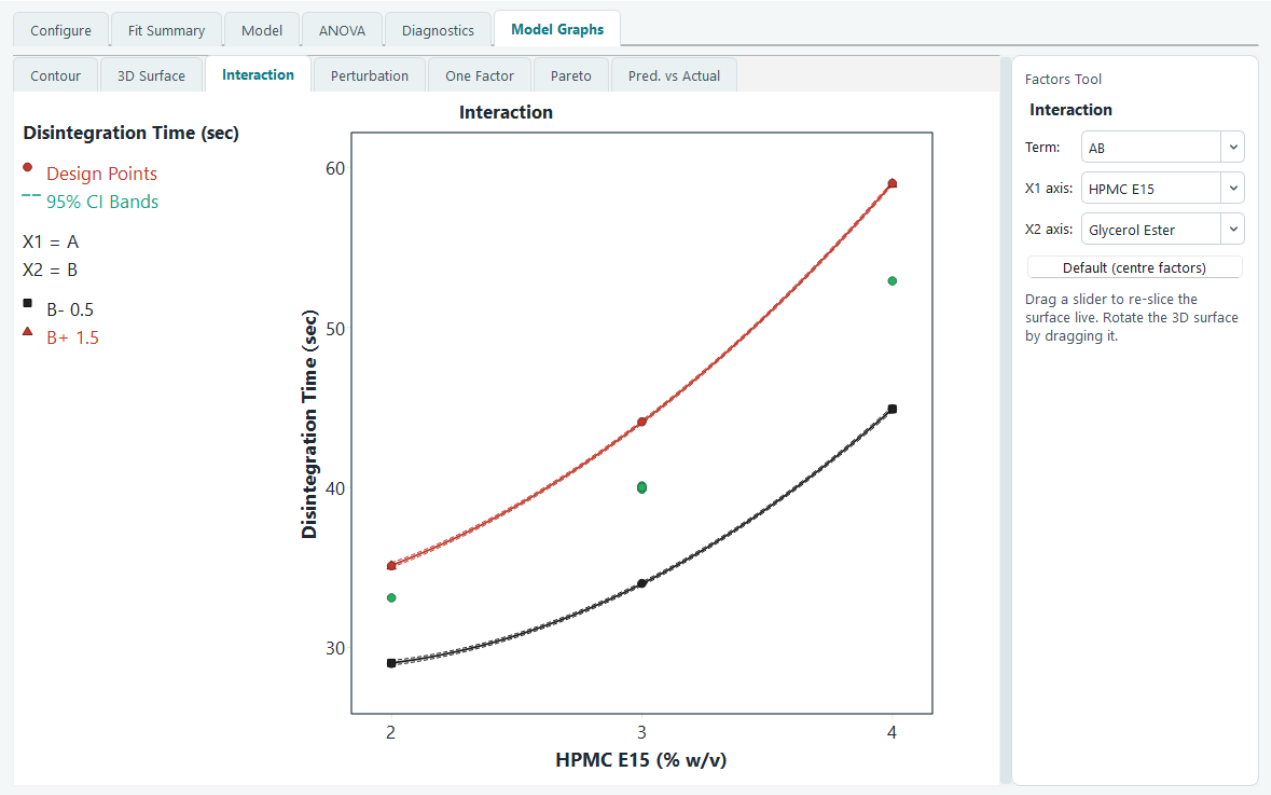
The fitted surface. Points above the surface are solid, points below are hollow, so you can see at a glance where the model over- or under-predicts.

Drag anywhere on the cube to rotate it. Dragging left and right spins it around; dragging up and down tilts it. The cube keeps a fixed size while you rotate, so it never jumps around under the cursor. To set an exact viewpoint instead of hunting for it, right-click the graph and choose *Set rotation*, then type a horizontal angle and a vertical angle. This is the way to make two surfaces line up for a side-by-side comparison, or to reproduce the same view every time.

> **Contour for numbers, surface for shape.**
> Read exact values off the contour map, where the labelled lines and the hover read-out give you figures. Use the 3D surface to explain the shape of the response to someone else. They are two views of one model, so they always agree.

Interaction

An interaction graph answers one question: does the effect of one factor depend on the level of another? It plots the predicted response against X1 as a line, drawn twice, once with X2 held at its low setting and once with X2 held high. If the two lines are parallel, the factors act independently. If they fan apart or cross, there is an interaction, and the further from parallel they are, the stronger it is.

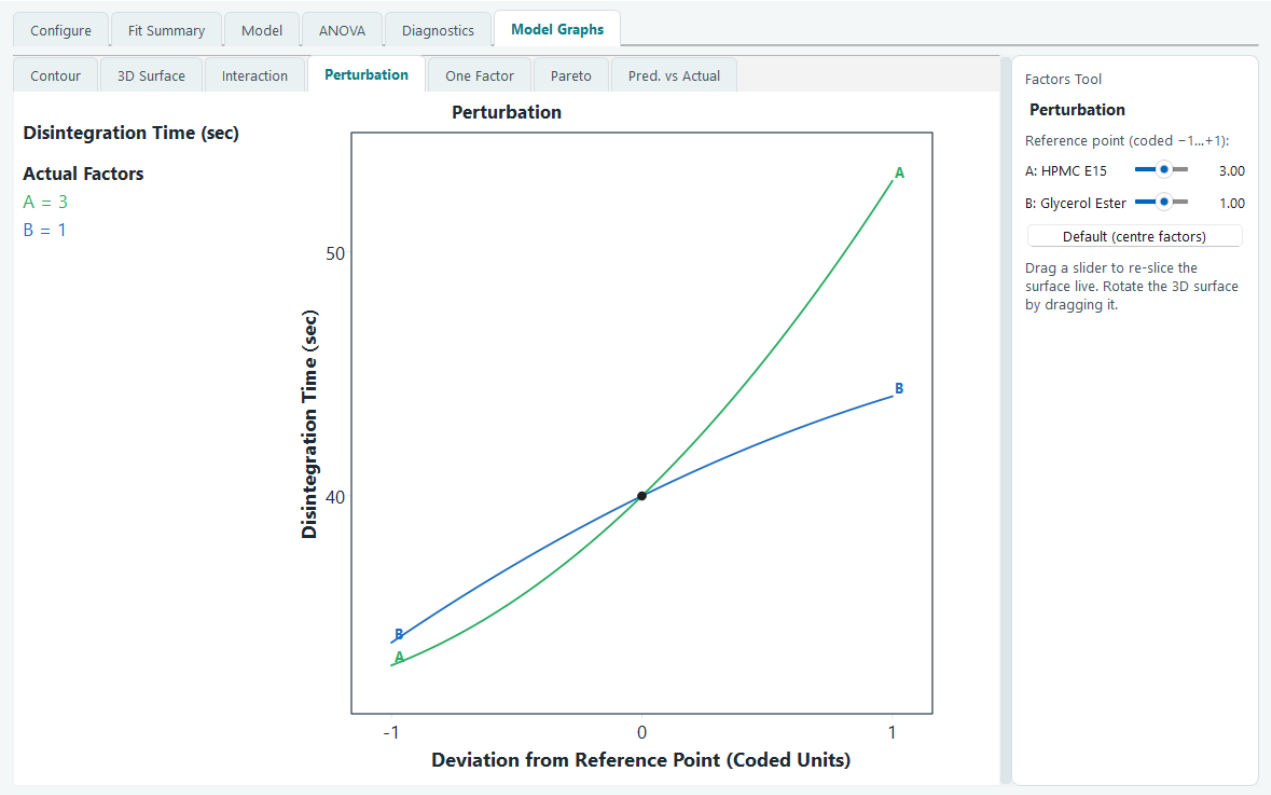


Interaction of A with B. The lower line holds B at 0.5, the upper line holds B at 1.5. Dashed curves above and below each line are the 95% confidence bands.

Each line carries a dashed **95% confidence band** above and below it. The band is the range the true mean response is expected to lie in, so where a band is wide the prediction is less certain, and where two bands overlap you cannot claim the two levels really differ at that point. Observed runs that lie on the current slice are plotted too, coloured by which level of the second factor they belong to, so you can see the data behind the lines.

Perturbation

The perturbation graph puts every factor on one plot so you can compare their influence at a single point. Starting from a reference point, it moves one factor at a time away from that point and draws how the response changes, giving one curve per factor. The horizontal axis is the coded deviation from the reference, so all factors share the same scale and are directly comparable.

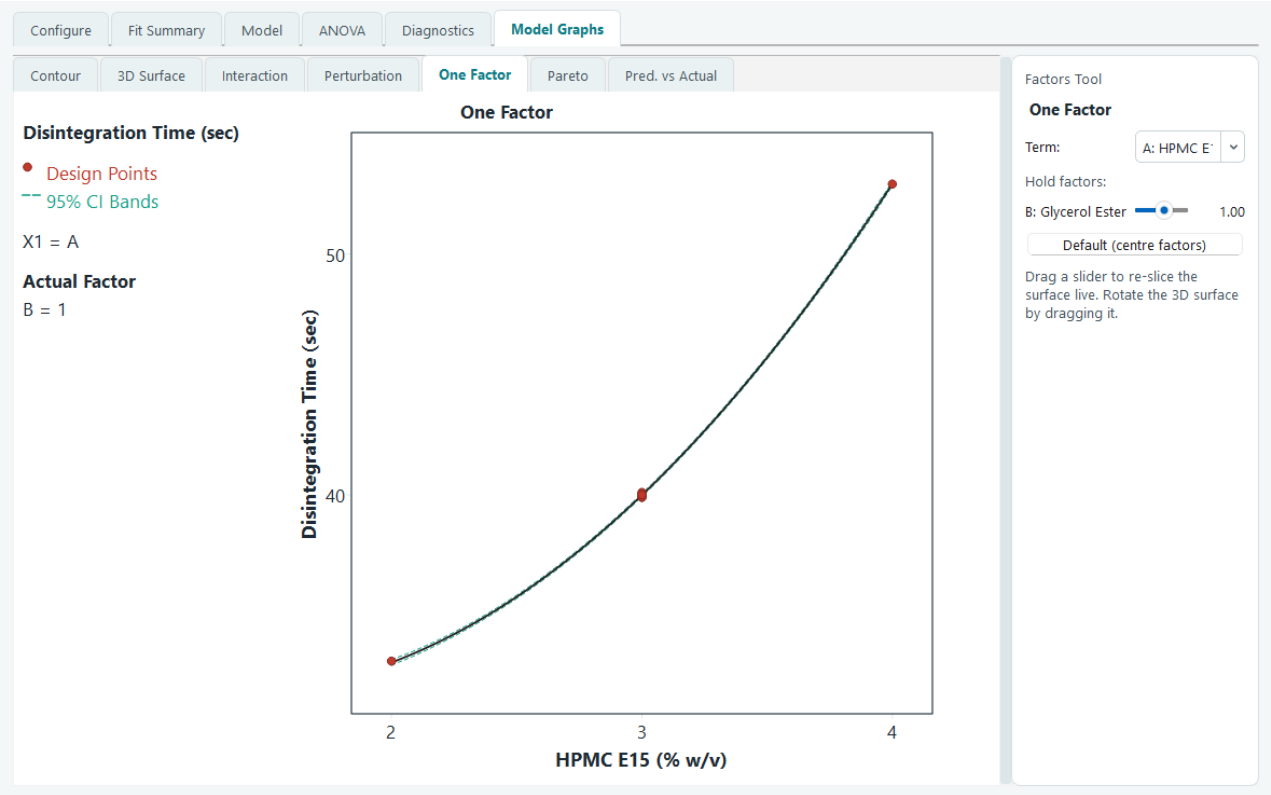


Perturbation from the centre point. Each curve is labelled with its factor letter at both ends; the black dot marks the reference point.

Read it by slope. A steep curve marks a factor the response is sensitive to; a flat curve marks one that barely moves it near this point. A curved rather than straight line signals curvature in that factor. The reference point is set by the sliders in the Factors Tool, which here are labelled as the reference rather than as held factors, so you can ask the same question at the centre, at a corner, or anywhere else worth checking. The black dot is the reference itself, where every curve meets.

One Factor

The One Factor graph is the simplest of the set: the predicted response against a single factor, with the other factors held wherever the sliders leave them. It is the right graph for a factor that the ANOVA showed has no meaningful interactions, because then this single line tells its whole story.

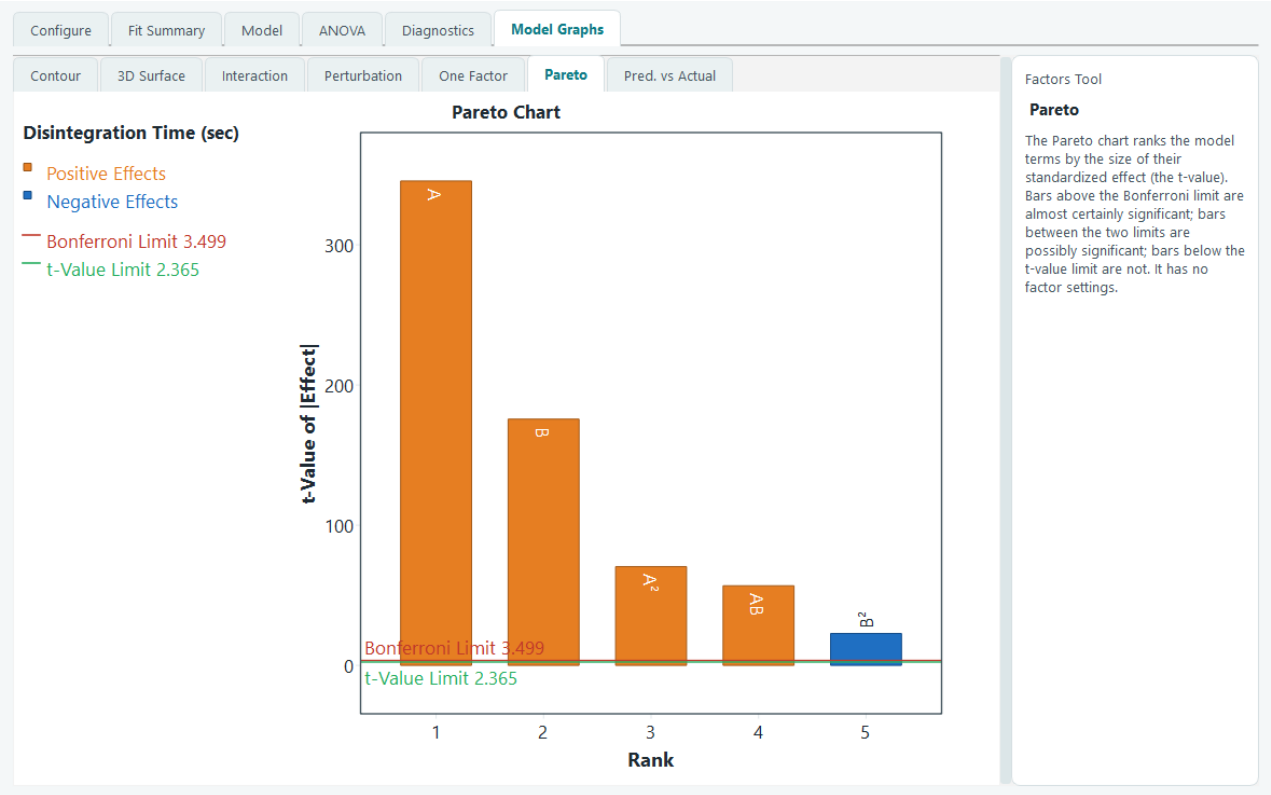


One Factor graph. The solid line is the fitted response, the dashed curves are the 95% confidence band, and the red dots are the runs that sit on the current slice.

As on the interaction graph, the dashed curves are the 95% confidence band around the fitted line, and the red dots are the observed runs that lie on the slice you are showing. If the points hug the line and stay inside the band, the model describes that factor well.

Pareto

The Pareto chart answers one question at a glance: which terms in the model actually matter? Every term is drawn as a bar whose height is the size of its standardized effect (its t-value), sorted from largest to smallest. Orange bars push the response up, blue bars pull it down, and the term name is written along each bar.



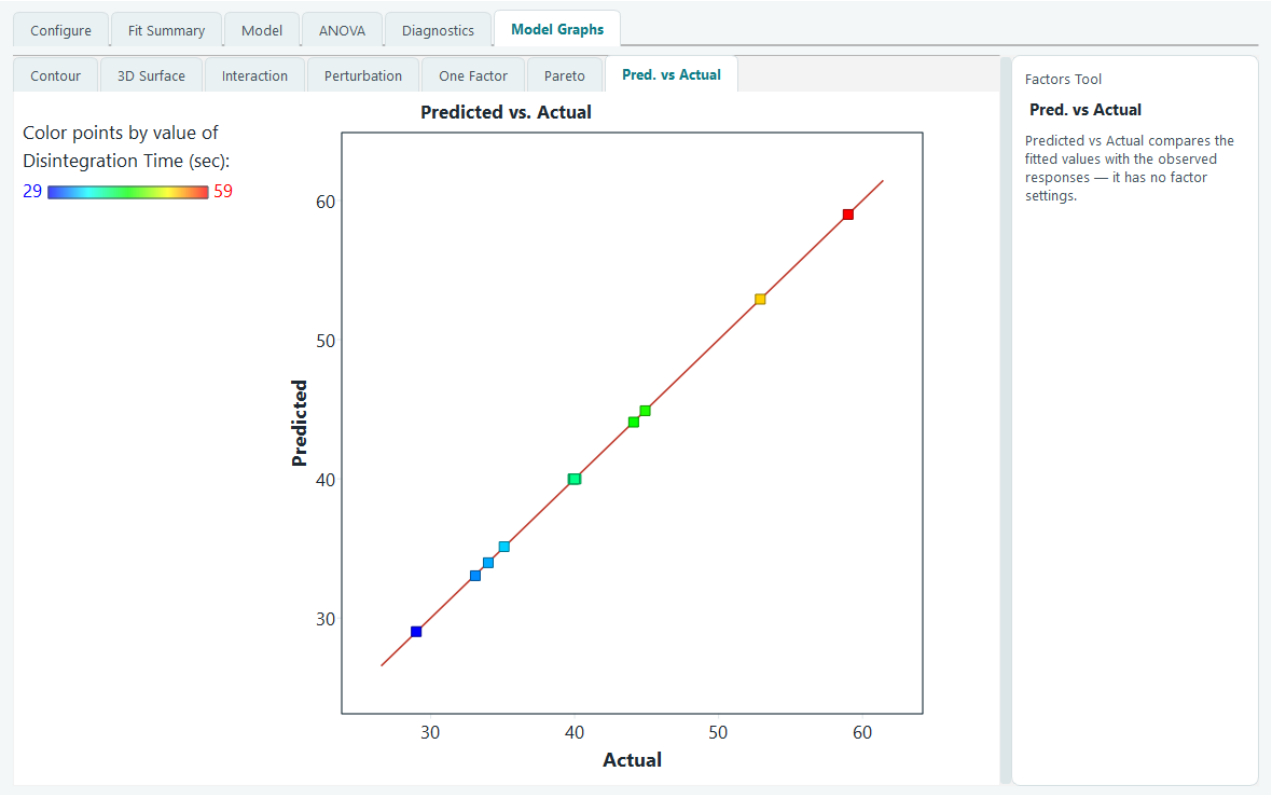
Pareto chart. Bars are the model terms ranked by the size of their standardized effect; the two horizontal lines are the significance limits.

The two horizontal lines are the reference points. A bar that clears the red **Bonferroni limit** is almost certainly a real effect even after allowing for the number of terms being tested at once. A bar between the two lines is possibly significant, and one below the green **t-value limit** shows no evidence of an effect. The exact limit values are listed in the left panel, and they match the p-values on the ANOVA page: this chart is the same test drawn as a picture.

The Pareto chart is offered for process designs, where every factor moves independently. Mixture designs do not show it, because the size of a blend component's coefficient depends on how the blend constraint is written into the model, so ranking those bars would mislead.

Predicted versus Actual

The last tab repeats the Predicted versus Actual scatter from the Diagnostics page, so you can keep a check on model quality without leaving the graphs. Each run is a point: its measured response along the bottom, its fitted value up the side. The diagonal line is where a perfect prediction would land.



Predicted versus actual. Points cluster along the diagonal when the model tracks the data; the colour of each point follows its measured value.

Points that sit tight to the line mean the model is tracking your data. A point that strays far from it is one the model predicts poorly, and is worth a look on the Diagnostics tab to see whether it is an outlier or a sign the model needs another term.

Copying and exporting any graph

Every graph here behaves the same way when you right-click it. You get **Copy**, which puts the picture on the clipboard ready to paste into a document, and an **Export** submenu that sends the graph straight into Word or PowerPoint, or saves it as a PNG or JPEG file. On the 3D surface the same menu also carries *Set rotation*.

The export is always readable.
A graph is exported at a comfortable size no matter how small its window is on screen. Before the image is captured the software re-lays the graph out at a generous size and renders it at high resolution, so axis titles are never clipped, the side panel never collides with the plot, and the picture stays sharp when it is enlarged in a report or a slide. You do not have to widen the window first to get a clean export.

With the model understood as a picture, the next step is to ask the software to find the settings that meet every response at once. That is what the optimization pages are for.

Part 5

Finding the best settings

Chapter 15

Numerical optimization

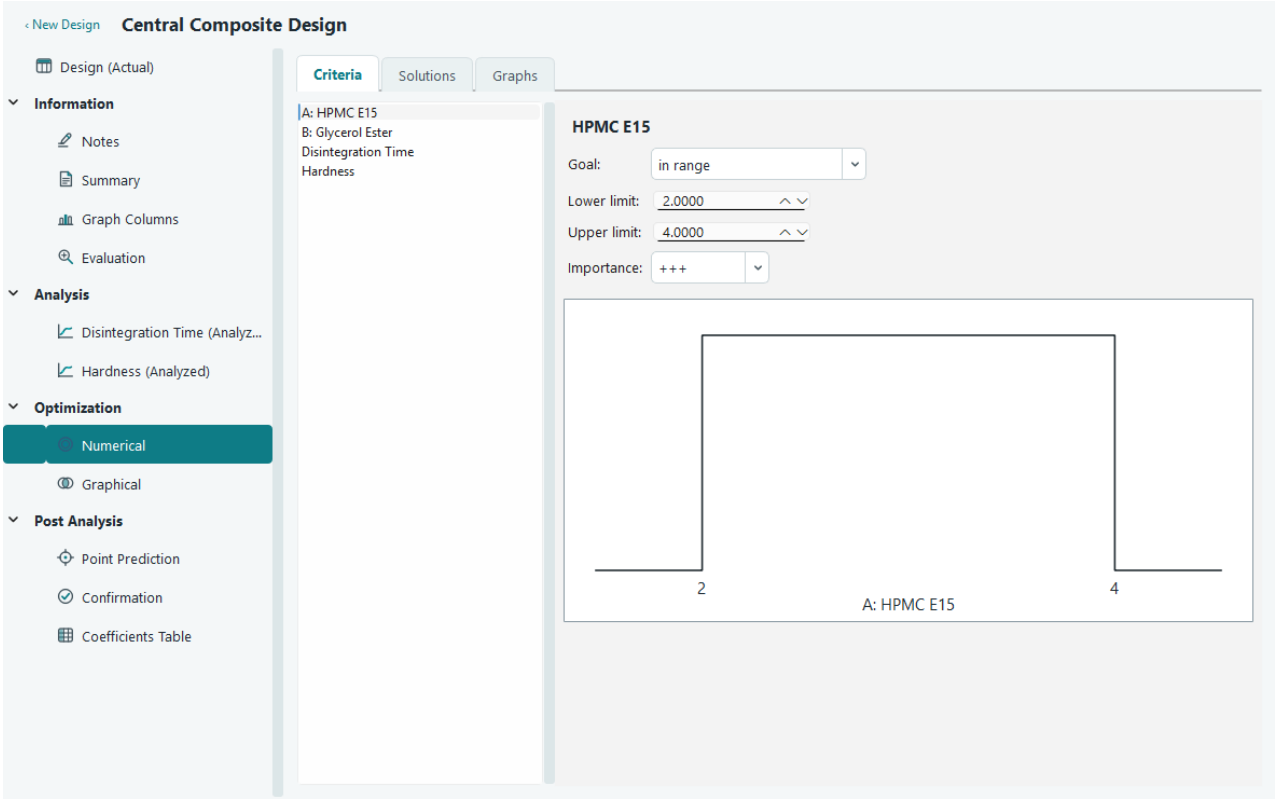
Most experiments have more than one response, and the settings that are best for one are rarely best for another. Numerical optimization asks the fitted models a single question: which factor settings come closest to satisfying every goal at once. It answers by turning each goal into a desirability score, combining those scores, and searching for the highest total.

You reach this screen from the Optimization branch of the navigation pane, under **Numerical**. It has three tabs that you work through in order: **Criteria**, where you say what you want; **Solutions**, where the software lists the settings it found; and **Graphs**, where you see those settings in context.

> **Analyse first.**
> Only responses you have already analysed take part. Each goal is measured against that response's fitted model, so if a response has no model yet, go back to its Analysis page and fit one before you come here.

Setting a goal for each factor and response

The Criteria tab lists your factors first, then your responses. Select one on the left and its editor appears on the right: a goal, a lower and upper limit, sometimes a target, weights, and an importance. Below the editor sits a picture of the desirability ramp for the current goal.



The Criteria tab. The ramp at the bottom shows how a result is scored between the lower and upper limits.

A goal is how you want a response to behave. Every response gets a desirability between 0 and 1: a 1 means the goal is fully met, a 0 means it is missed completely, and values in between measure how close you are.

Goal	For a response	For a factor
none	Predict and report it, but leave it out of the search.	Not offered.
maximize	Score rises from 0 at the lower limit to 1 at the upper limit.	Pin the factor at its upper limit.
minimize	Score rises from 0 at the upper limit to 1 at the lower limit.	Pin the factor at its lower limit.
target	Score peaks at the target and falls to 0 at each limit.	Pin the factor at the target value.
in range	Any value between the limits scores 1, anything outside scores 0.	Let the search move the factor freely between the limits.
equal to	Not offered.	Pin the factor at the value.

Factor goals do not add to the desirability. They shape where the search is allowed to look. Leave a factor *in range* and the optimizer is free to tune it; give it any other goal and you fix it, so you can ask questions like "what is the best hardness I can get if HPMC E15 is held at 3 % w/v".

For the worked example in these pages, a face-centred central composite design on A: HPMC E15 and B: Glycerol Ester, you might minimize Disintegration Time and aim Hardness at a target, while leaving both factors in range so the search can move them.

The lower and upper limits

The limits are the two ends of the ramp. For a response they default to the smallest and largest values you actually measured, which is a sensible starting range. Widen or narrow them to match what you care about. A result below the lower limit or above the upper limit scores 0 for that response, so the limits are also a hard boundary, not just a scale.

Weights and the shape of the ramp

A weight bends the ramp between a limit and the good end. With a weight of 1 the ramp is a straight line, so desirability grows evenly. A weight above 1 bends the ramp downward, so a result earns a high score only when it is close to the good limit. A weight below 1 bends it upward, so even a partial result already counts for a lot. There is a lower weight for the ramp on the low side and an upper weight for the high side; a target goal uses both, one for each slope.

You can set the weights in two ways. Type them into the Lower weight and Upper weight boxes, which accept values from 0.1 to 10, or drag the ramp directly. Each sloped part of the ramp has a small square handle at its middle. Drag a handle up to soften the ramp, which lowers the weight, or drag it down to make it stricter, which raises the weight. The boxes follow the handle, and the handle follows the boxes, so the two always agree.

Start straight.
Leave weights at 1 unless you have a reason to change them. Bend a ramp only when a response matters much more near one end, for example when a tablet must be well above a minimum hardness rather than merely above it.

Importance

Importance says how much a response counts against the others. You set it from one plus (+) to five pluses (+++++), with three pluses in the middle as the default. A response with five pluses pulls the search toward its own goal much harder than one with a single plus. Use it to break ties between responses that cannot all be satisfied perfectly.

How the scores combine

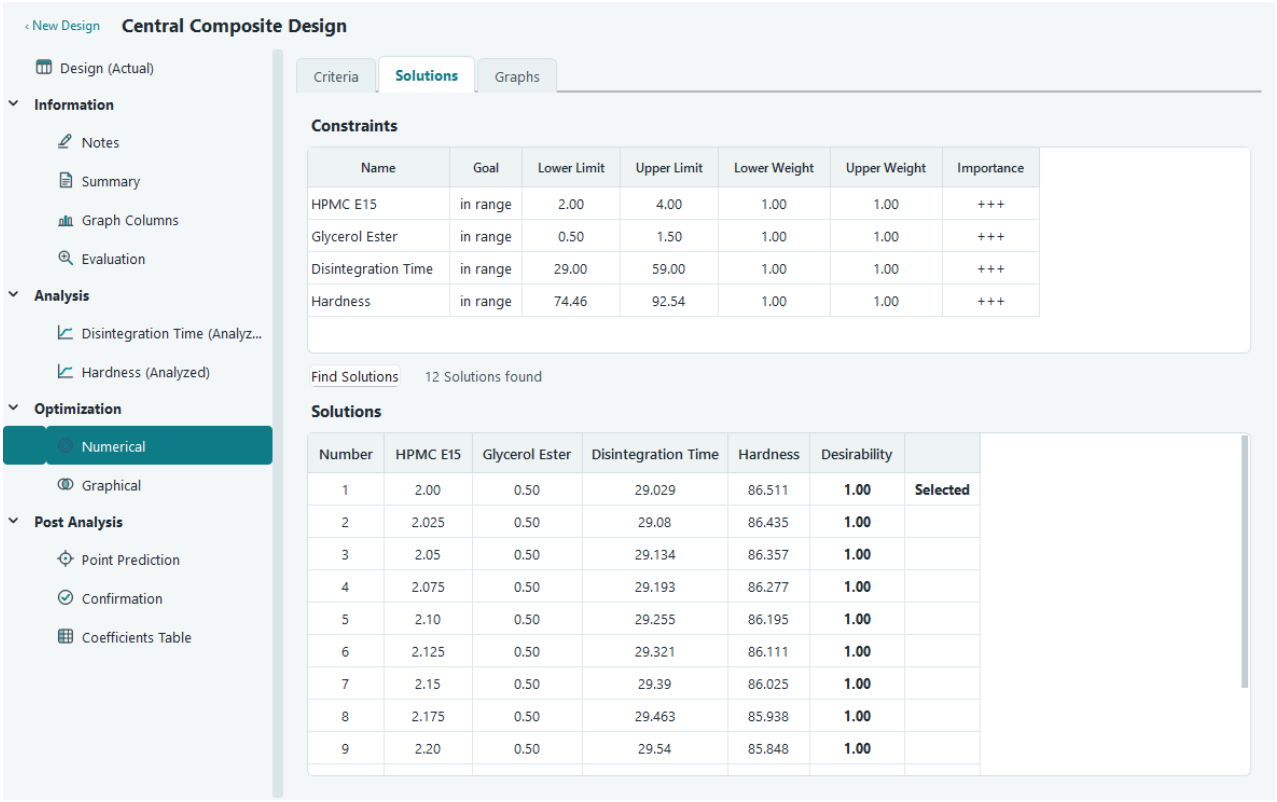
Each response gives one desirability. The software combines them into a single overall desirability, written as D, using an importance-weighted geometric mean. In plain terms, it multiplies the scores together, giving more say to the responses you marked as more important, then takes the matching root. Responses set to *none* are left out of this sum.

One zero sinks everything.
Because the scores are multiplied, a desirability of 0 anywhere forces the overall D to 0. A single response outside its limits makes the whole solution unacceptable, no matter how well the others do. If every solution comes back at 0, a goal or a limit is too strict to reach.

Desirability is a search tool, not a grade.
A D of 1.00 means only that every goal you wrote was met. It does not mean the result is good in any real sense. If your limits were loose or your goals modest, a perfect score is easy and tells you little. Read the predicted response values, not just D, and judge them against what you know.

The Solutions table

Open the Solutions tab and the search runs. At the top, a Constraints table repeats every goal, limit, weight and importance you set, so the run is self-documenting. Press **Find Solutions** to run it again after any change.

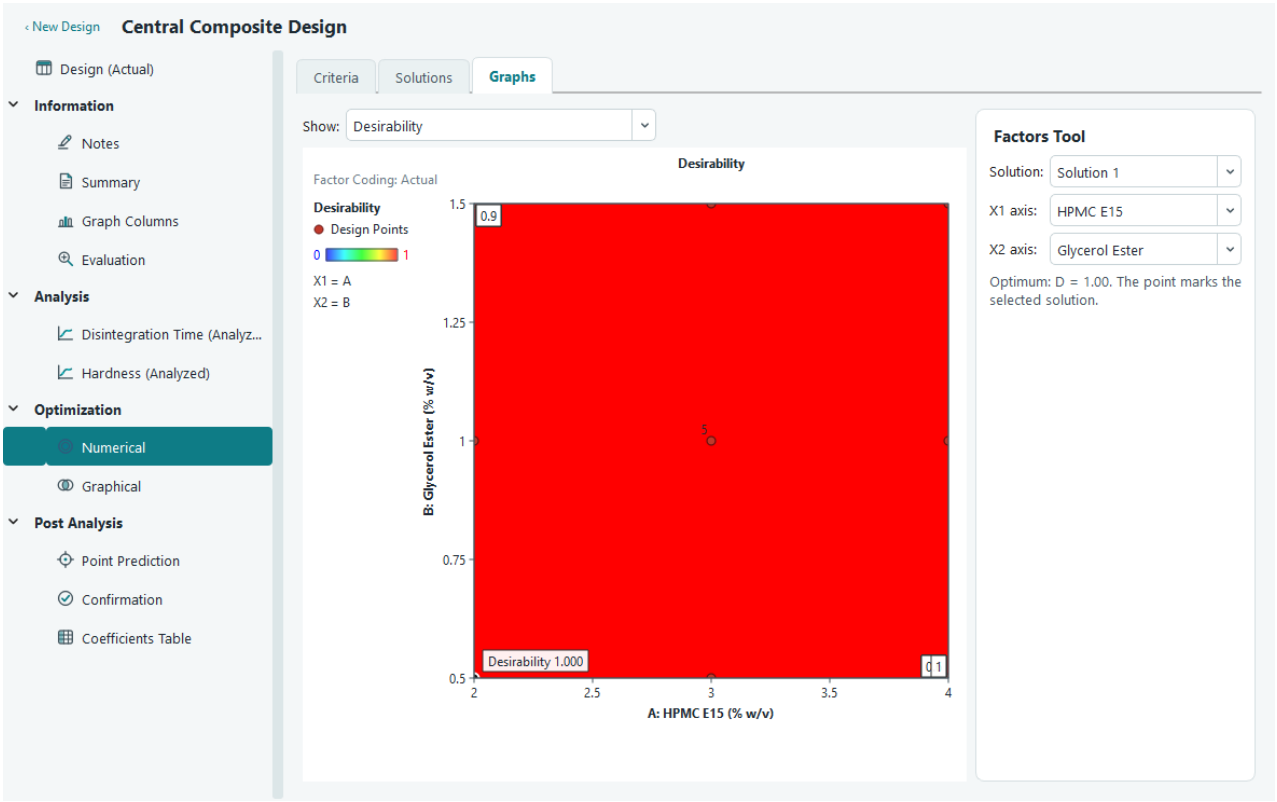


Solutions, best first. Each row is one set of factor settings, with the predicted responses and the overall desirability.

Each row is a set of factor settings the optimizer found, with the values it predicts for every response and the overall desirability for that row. The list is sorted with the highest desirability first, and the top row is marked Selected. The optimizer reports up to a dozen distinct settings, so you can compare near-equal options rather than trusting a single answer. Two rows with almost the same desirability may sit in very different places, and one may be far easier to run than the other.

The Graphs tab

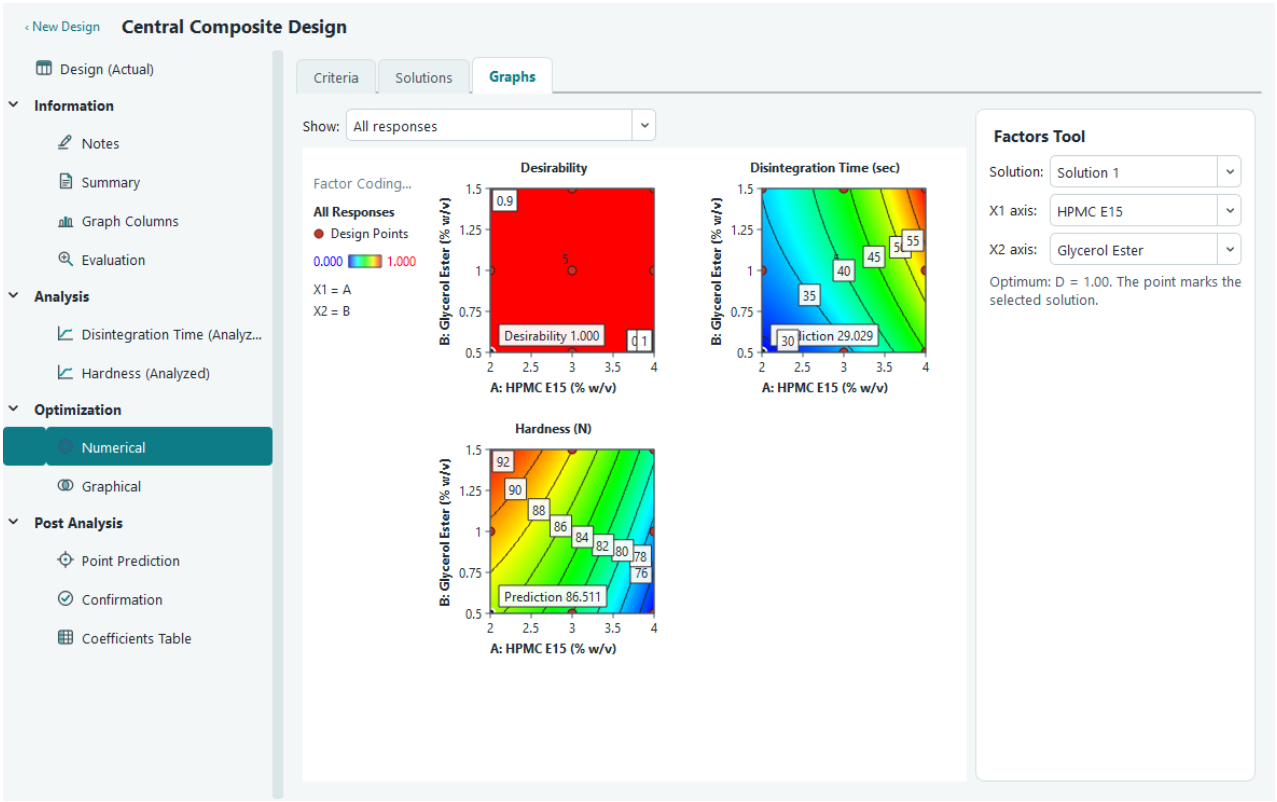
Numbers in a table are hard to trust on their own. The Graphs tab draws the solution on the fitted surfaces so you can see how forgiving it is. Pick which surface to show with the **Show** list: the overall desirability, any single response, or **All responses** at once. On the right, the Factors Tool lets you choose which solution to plot and which two factors go on the axes.



The Graphs tab. The marker sits at the selected solution; a broad flat region around it means the setting is robust.

The desirability map is worth a look before you commit. A solution sitting on a steep slope or a narrow ridge is fragile: a small drift in a factor sends desirability down fast. A solution in a wide, gentle plateau is one you can hit in practice. The chosen point is marked on every plot, labelled with its overall desirability or its predicted value.

The **All responses** view puts the desirability map and one map per response into a single figure with a shared legend panel. The maps fill as many columns as your window allows and wrap onto new rows below when the width runs out, so every map stays a readable size whether you have two responses or ten, on a laptop screen or a large monitor. It is the clearest single picture of a solution, showing at a glance where each response lands as you move around the two factors. Because it is drawn as one figure, copying or exporting it gives you one combined image for a report.



The All responses view. Maps share one legend panel and wrap onto extra rows when the window is narrow.

Picking a solution and confirming it

The top solution is a suggestion, not a verdict. Choose the one you will actually run, then check it in the lab.

1. On the Solutions tab, read across the rows and pick the settings that best fit your process. Click that row to select it. The Graphs tab and the Factors Tool follow your choice.
2. Switch to the Graphs tab and confirm the point sits in a stable region of the desirability map, not on a steep edge.
3. Note the factor settings from that row. Use Point Prediction under Post Analysis to read the predicted responses at those settings, together with their confidence and tolerance intervals.
4. Run the chosen settings in the lab, then enter the measured responses on the Confirmation screen. It compares what you got against what the model predicted, so you can trust the optimum before you rely on it.

> **Optimization does not change your data.**
> The search only reads the fitted models. Nothing here edits the run sheet, so you can try different goals, weights and importances freely and press Find Solutions as often as you like.

Chapter 16

Graphical optimization

Numerical optimization hands you a single best set of factor values. Graphical optimization shows you the whole neighbourhood around it: the range of settings where every response still meets its requirements. Use the two together and you get both the point and the room around it.

Why look at a picture

A single recommended point is easy to act on but tells you nothing about how forgiving it is. In the factory or the lab, factors drift. HPMC concentration is a little off, the room is warmer than usual, a batch of glycerol ester behaves slightly differently. What you really want to know is how much slack you have before a response falls out of specification.

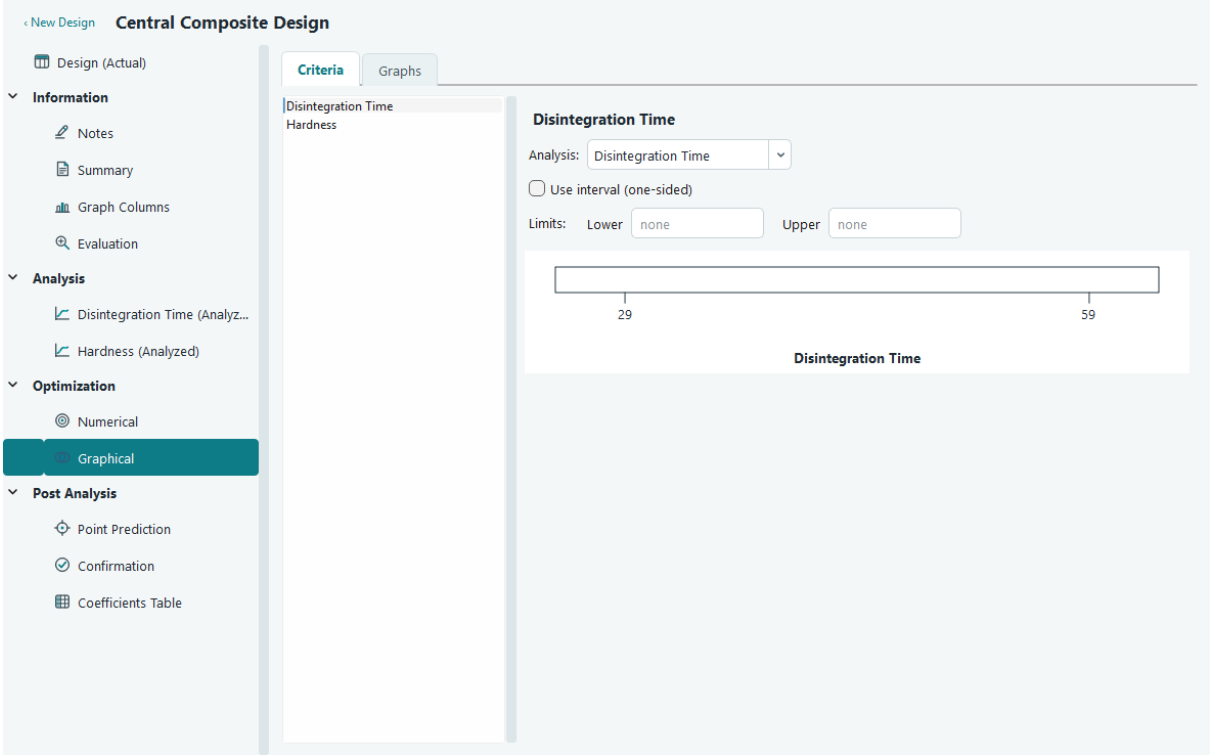
The overlay plot answers that. You give each response a lower limit, an upper limit, or both. OptiGein Design then colours the factor space: yellow where every response stays inside all of its limits, grey where at least one response fails. The yellow patch is your operating window. A large patch means a robust process; a thin sliver means you have almost no margin, even if the numerical optimum sits inside it.

> **Do numerical first.**
> Run Numerical optimization to find where the sweet spot is, then open graphical optimization to see how big it is. The point tells you where to aim; the picture tells you how much you can miss by.

Set the limits

Open **Optimization › Graphical** in the navigation pane. It has two tabs, **Criteria** and **Graphs**. Start on Criteria. The list on the left shows every response you have already analysed. A response only appears here once it has a fitted model, so if one is missing, go back to its Analysis page and fit it first.

1. Select a response in the list. The panel on the right names it and shows the range actually observed in your runs, so you have a sensible frame of reference for the numbers you are about to type.
2. Type a value in **Lower** , in **Upper** , or in both. Leave a box empty (it reads *none*) to say that side has no limit. For the worked example you might cap Disintegration Time (sec) with an upper limit, because a tablet that takes too long to disintegrate is no good, and give Hardness (N) a lower limit, because a tablet that is too soft will crumble.



The Criteria tab. Each response gets its own lower and upper limits. The strip below shows the observed range with the chosen limits marked in red and the in-limit band shaded yellow.

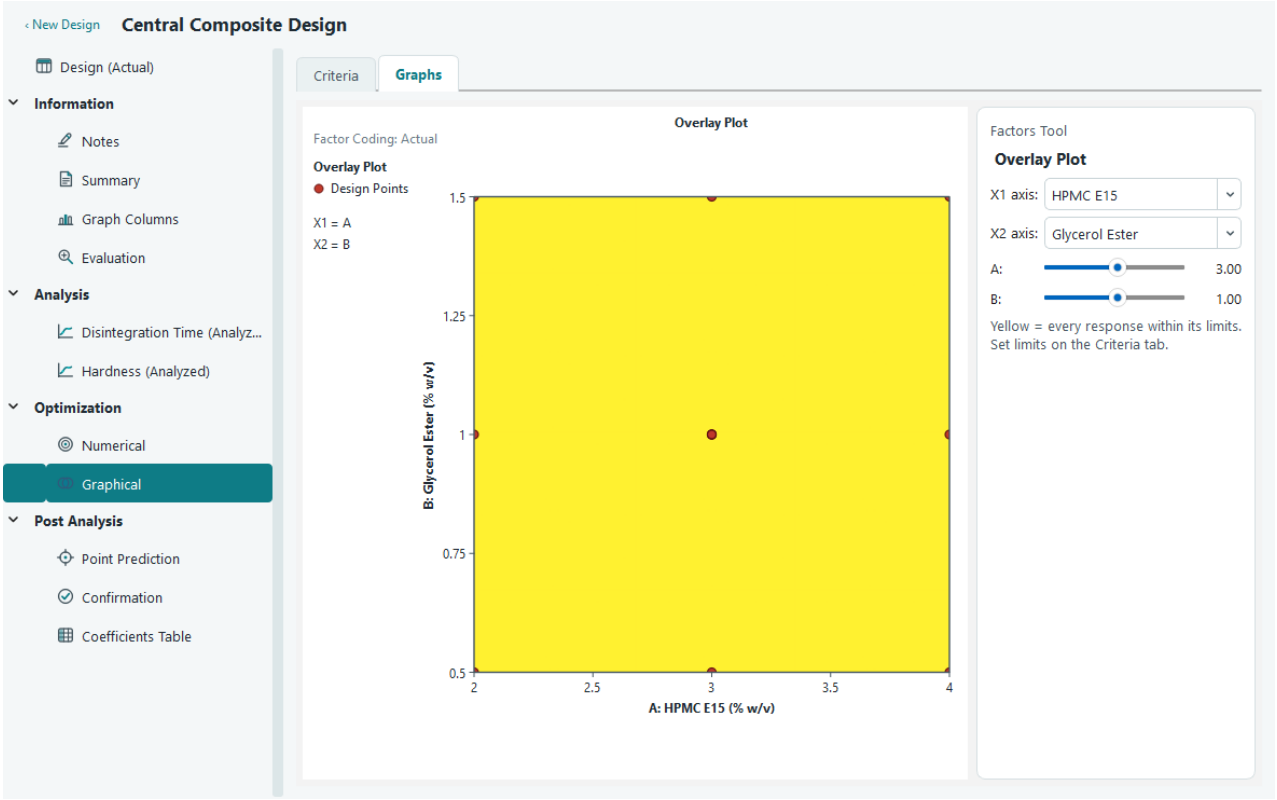
3 Repeat for every response you care about. A response with no limits set does not constrain the plot at all, so only fill in the ones that matter.

The strip beneath the fields is a quick sanity check. It draws the response range your experiment actually produced, marks your lower limit with a ≥ and your upper limit with a ≤, and shades the band between them yellow. If your limits sit right at the edge of the observed range, or outside it, that is worth noticing before you read the overlay. If you type a lower value above an upper value by mistake, the software quietly swaps them so the band still makes sense.

Predicted mean, or a confidence bound.
By default the yellow region is judged from the predicted mean of each response. Tick *Use interval (one-sided)* and OptiGein Design instead requires the response to stay inside its limit with 95% confidence, using the one-sided confidence bound on the mean. That is a stricter test, so the yellow region shrinks. It is the honest choice when you want settings that will hold up in practice, not just settings where the best guess happens to be in spec.

Read the overlay

Switch to the **Graphs** tab. The overlay plot fills most of the page, with a **Factors Tool** panel on the right. Two axes carry two of your factors; every other factor is held fixed. For the two factor example, A: HPMC E15 and B: Glycerol Ester go straight onto the axes and nothing is held.



The overlay plot for the worked example. Yellow is the operating window where both Disintegration Time and Hardness meet their limits; grey is out of spec. The coloured curves are the individual limit boundaries, and the red dots are your design runs.

What you see	What it means
Yellow region	Every response stays inside every limit you set. Any settings in here satisfy all of your requirements at once.
Grey region	At least one response fails at least one limit. You do not have to know which one; grey simply means out of bounds.
Coloured boundary curve	The line where one response is exactly equal to one of its limits. Cross it and that response tips over the edge. Each active limit gets its own colour, and the same colour keys the legend in the info panel on the left.
Red dots	Your actual design runs that lie in the plane being shown. They tell you whether your operating window is near real data or out in extrapolated territory.

Each boundary curve carries a small label naming the response and the limit value, so you can tell at a glance which edge of the yellow region belongs to which requirement. The left panel lists the same limits with their colour swatches, the factors on the two axes, and the fixed value of any held factor.

Hover to read exact numbers.
Move the pointer anywhere inside the plot and a small read-out follows the cursor, showing the factor values at that spot and the predicted value of every response there. It is the quickest way to check how much headroom a candidate setting has before it reaches a boundary.

When you have more than two factors

You can only draw two factors on the axes, so a design with three or more factors is shown as a slice: a flat cut through the factor space at fixed values of everything not on the axes. The Factors Tool panel gives you an **X1 axis** and **X2 axis** selector to choose which two factors to plot, and a slider for each remaining factor to set where the slice is taken. Its actual value updates beside the slider as you drag.

Moving a slider re-cuts the plane, and the yellow region grows, shrinks or shifts as you go. This is the point of graphical optimization with many factors: the operating window is not one fixed shape, it depends on where the held factors sit. Slide through them to find the combination that gives you the most generous window. The red design points also change, because only the runs that actually lie in the current slice are drawn.

When there is no yellow region

Sometimes the whole plot is grey. That is information, not a failure. It means no setting in the region you are looking at can satisfy all of your limits together. Before you give up, try these in order.

An empty window means your targets are fighting each other.
Two responses often pull in opposite directions, so tightening both at once can leave nothing that meets everything.

- **Loosen a limit.** Go back to the Criteria tab and relax the requirement that is hardest to meet. Even a small change can open a window. Watch the range strip to keep your limits inside what the data can actually deliver.
- **Turn off the confidence test.** If *Use interval (one-sided)* is ticked, the region is being drawn conservatively. Clearing it judges by the predicted mean instead, which is more forgiving and may reveal a window that is genuinely there but marginal.
- **Move the slice.** With more than two factors, an empty window on one slice can be a healthy window on another. Drag the held-factor sliders and watch for yellow to appear.
- **Reconsider the goals.** If nothing helps, your requirements may simply be beyond what this system can do. That is a real and useful finding. Check what the numerical optimizer reports too: if its best desirability is low, the two tools are telling you the same thing.

Keep the picture.
Right-click the overlay plot to copy it or export it as an image. It is a clear way to show a colleague or a report exactly where the safe operating window sits.

Chapter 17

Prediction and confirmation

The Post Analysis branch is where you use the models you built. Read off a predicted response at any settings you like, understand how sure the software is about that number, plan how many verification runs to perform, and compare all your response models side by side.

Once you have fitted a model to each response and looked at the diagnostics, the maths is done. Post Analysis turns those models into answers you can act on. It has three pages, and each one answers a different practical question.

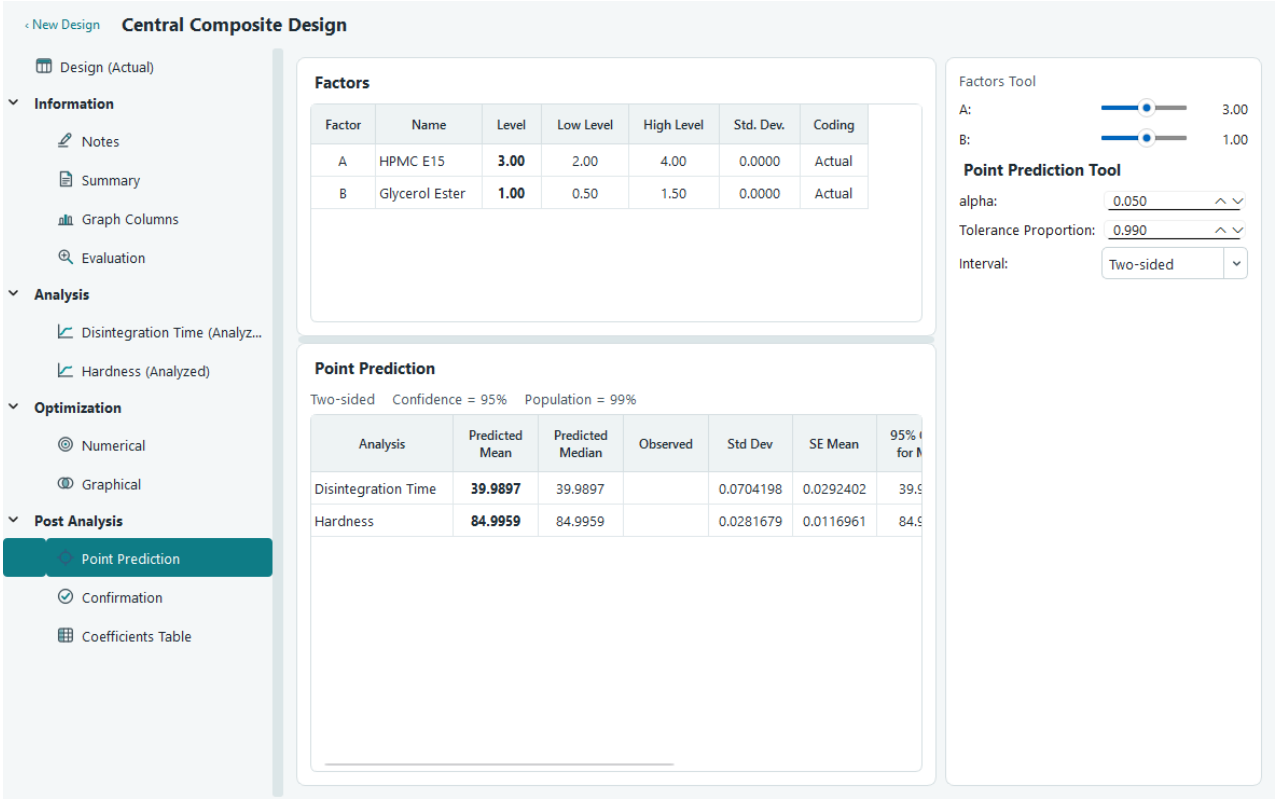
Page	The question it answers
Point Prediction	What response should I expect at these exact settings, and how much could it vary?
Confirmation	I ran the recipe a few times. Do my results agree with the model?
Coefficients Table	How do all my response models compare, term by term?

Only fitted responses appear.
Every Post Analysis page reads the models from the Analysis branch. A response shows up here only after you have analysed it and the fit succeeded. If a response is missing from a table, go back to its Analysis node and complete the fit first.

Point Prediction

Point Prediction lets you pick any point inside the design space and read off what every response is expected to be there. In the worked example, a face-centred central composite design on `A:HPMC E15 (% w/v)` and `B:Glycerol Ester (% w/v)`, you might ask what Disintegration Time and Hardness the models predict at 3.2 % HPMC and 0.9 % glycerol ester, a setting you never actually ran.

The right-hand **Factors Tool** has one slider per numeric factor. Each slider moves that factor across its range, from its low level to its high level, and the label beside it shows the setting in real units as you drag. The Factors table at the top left echoes the chosen level for each factor next to its low and high values. The Point Prediction table below updates the moment you move a slider.



Point Prediction. Move the sliders on the right to choose a point; every fitted response is predicted at that point with its intervals.

Each row of the Point Prediction table is one response. The columns tell you the prediction and, just as important, how much to trust it.

Column	What it means
Predicted Mean	The model's estimate of the average response at this point. This is the number you would quote.
Predicted Median	The predicted middle value. For a response you did not transform, the median equals the mean, so the two columns match.
Std Dev	The estimated run to run standard deviation of the response about the model. It comes from the residual mean square and does not change as you move the point.
SE Mean	The standard error of the predicted mean. This measures how precisely the model pins down the average at this point. It is smallest near the centre of the design and grows towards the edges and corners.
CI low / CI high for Mean	The confidence interval for the average response at this point.
TI low / TI high for Pop	The tolerance interval, which brackets a stated proportion of individual future units.

Notice that *Std Dev* and *SE Mean* are different things and people mix them up. *Std Dev* describes how much a single batch scatters around the true value. *SE Mean* describes how unsure the model is about where the true average sits, and that depends on where you are in the design: predict near the centre and the software is confident, predict out in a corner where you have little data and it is less confident.

The three controls

Below the sliders, the **Point Prediction Tool** has three settings that shape the intervals.

Control	Effect
alpha	Sets the confidence level. The level shown is (1 minus alpha). The default 0.05 gives 95 % intervals. Lower it for wider, more cautious intervals.
Tolerance Proportion	The fraction of the population the tolerance interval must cover. The default 0.99 asks the interval to contain 99 % of future units.
Interval	Two-sided gives a low and a high bound. One-sided is for when you only care about one direction, for example a minimum acceptable hardness.

The header line above the table restates your choices in words, for instance *Two-sided, Confidence = 95 %, Population = 99 %*, so the interval columns are never ambiguous.

Confidence interval or tolerance interval?

This is the distinction that trips people up, so it is worth being exact.

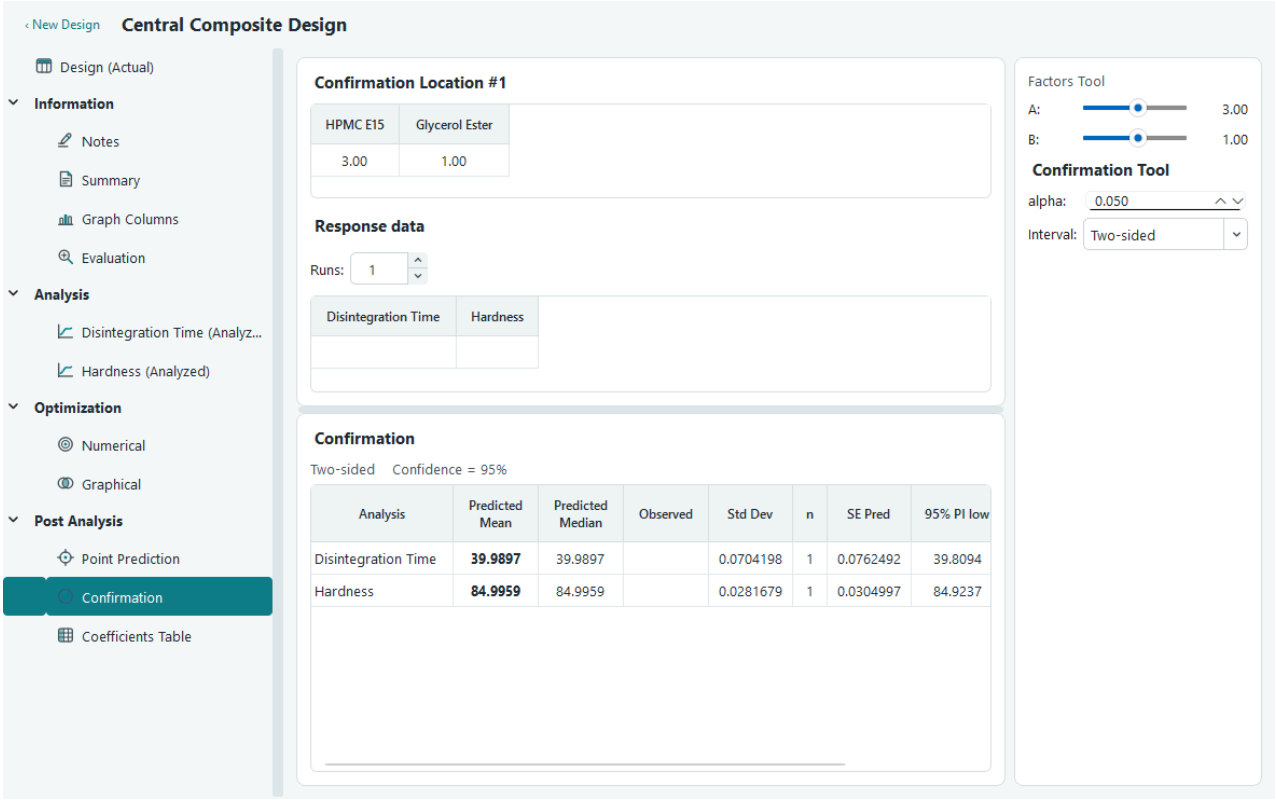
Interval	What it is a statement about
Confidence interval for the mean	A range that, with your chosen confidence, contains the *true average* response at this point. It is narrow, because averages are easier to pin down than individuals. It shrinks as your design gives more information about the mean.
Tolerance interval for the population	A range that, with your chosen confidence, contains a stated proportion of *individual future units*. It is wider, because it must allow for the run to run scatter of single batches, not just the position of the average.

> **Which one to quote.**
> Reporting the average behaviour of your process? Use the confidence interval for the mean. Promising that most individual batches will fall inside some spec? Use the tolerance interval for the population. The tolerance interval is always the wider of the two.

Confirmation

A model is a prediction until you test it. The Confirmation page helps you plan verification runs at a chosen setting and then check whether what you measured agrees with what the model expected. This closes the loop on the whole experiment.

The layout mirrors Point Prediction. The **Factors Tool** sliders set the location you want to confirm, shown as *Confirmation Location #1* with the factor settings in real units. The difference is the **Runs** box and the **Response data** table underneath it.



Confirmation. Set how many verification runs you plan, then type in what you measured and compare the average against the prediction interval.

1. Move the sliders to the setting you intend to verify. This is usually the optimum you found under the Optimization branch.
2. Set **Runs** to the number of confirmation experiments you plan to perform. This is the *n* column in the results. Raising *n* narrows the prediction interval, because the average of several runs is steadier than a single run.
3. Perform the runs, then type each measured value into the **Response data** table. Each analysed response gets its own column.
4. Read the **Data Mean** against the prediction interval. If the average of your confirmation runs lands inside the interval, your results agree with the model.

The Confirmation table gives the prediction for each response at the chosen point.

Column	What it means
Predicted Mean, Predicted Median, Std Dev	The same predictions as on the Point Prediction page.
n	The number of confirmation runs you set. It feeds directly into the interval.
SE Pred	The standard error of the mean of *n* future runs. It blends the run to run scatter with the model's own uncertainty, so it is larger than SE Mean and shrinks as you plan more runs.
PI low / PI high	The prediction interval. The average of *n* future runs at this point is expected to fall in this range.
Data Mean	The average of the values you typed into the Response data table. This is your evidence.

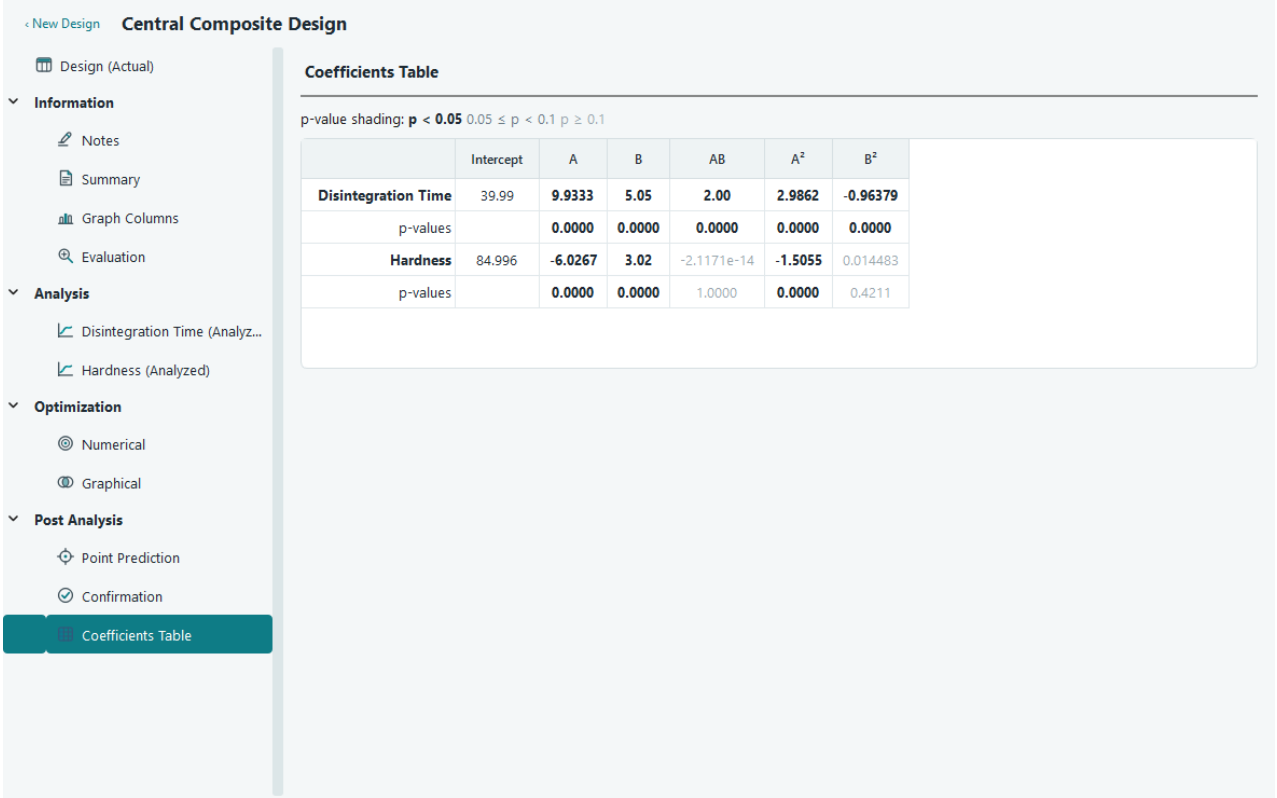
> **A prediction interval is not a confidence interval.**
> The confidence interval on the Point Prediction page describes the true average and ignores single run scatter. The prediction interval here describes a future *measured* result and therefore includes that scatter, which is why it is wider. Compare your confirmation runs against this prediction interval, not against the confidence interval.

If the Data Mean sits comfortably inside the prediction interval, the model has passed its test at that point. If it lands outside, something has drifted since you ran the design, or the model is missing an effect, and it is worth investigating before you rely on the settings.

The **alpha** and **Interval** controls behave as they do on the Point Prediction page. There is no population proportion here, because a prediction interval is about the average of your runs rather than a proportion of the population.

Coefficients Table

The Coefficients Table puts every fitted response model into a single grid so you can compare them at a glance. Each column is a model term, and every response contributes two rows: its coded coefficients and, beneath them, the p-values for those terms.



	Intercept	A	B	AB	A^2	B^2
Disintegration Time	39.99	**9.9333**	**5.05**	**2.00**	**2.9862**	**-0.96379**
p-values		**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Hardness	84.996	**-6.0267**	**3.02**	-2.1171e-14	**-1.5055**	0.014483
p-values		**0.0000**	**0.0000**	1.0000	**0.0000**	0.4211

The Coefficients Table. Two rows per response, the coded coefficients above and the p-values below, with significant terms shown darker.

The columns are the pooled list of terms across all your models, ordered from the largest model, so if Disintegration Time carries a curvature term that Hardness does not, that term still gets a column and Hardness simply leaves it blank. A blank cell means the term is not in that response's model.

The p-values are shaded so the important terms stand out without you reading every number. Darker, bold values are significant at p below 0.05. A medium shade marks borderline terms between 0.05 and 0.1. The lightest shade is for terms with p at or above 0.1. The intercept has no p-value, so its cell in the p-value row is left empty.

What to look for.
Read across a row to see which terms drive a response. Read down a column to see whether the same factor matters for every response or only for some. This is the fastest way to spot that one factor, say HPMC, controls both responses while another only moves one of them.

The coefficients here are in coded units, the same values you see in the coded equation on the Model tab of the Analysis branch. Coded coefficients are directly comparable in size, so a larger coefficient means a stronger effect over the factor's range, which is exactly what makes this table useful for comparison.

Where this fits

Post Analysis is the end of the journey: you built a design, filled in results, fitted and checked models, and searched for the best settings. Here you read predictions at any point, verify them with real runs, and compare your models. The natural next step is to save, print or export what you found.

- Numerical optimization to find the settings you would bring here to confirm.
- Saving, printing and exporting to get these tables into a report. Right-click any table to copy it.
- Glossary for short definitions of confidence, prediction and tolerance intervals.

Part 6

Tutorials

Chapter 18

Response surface, start to finish

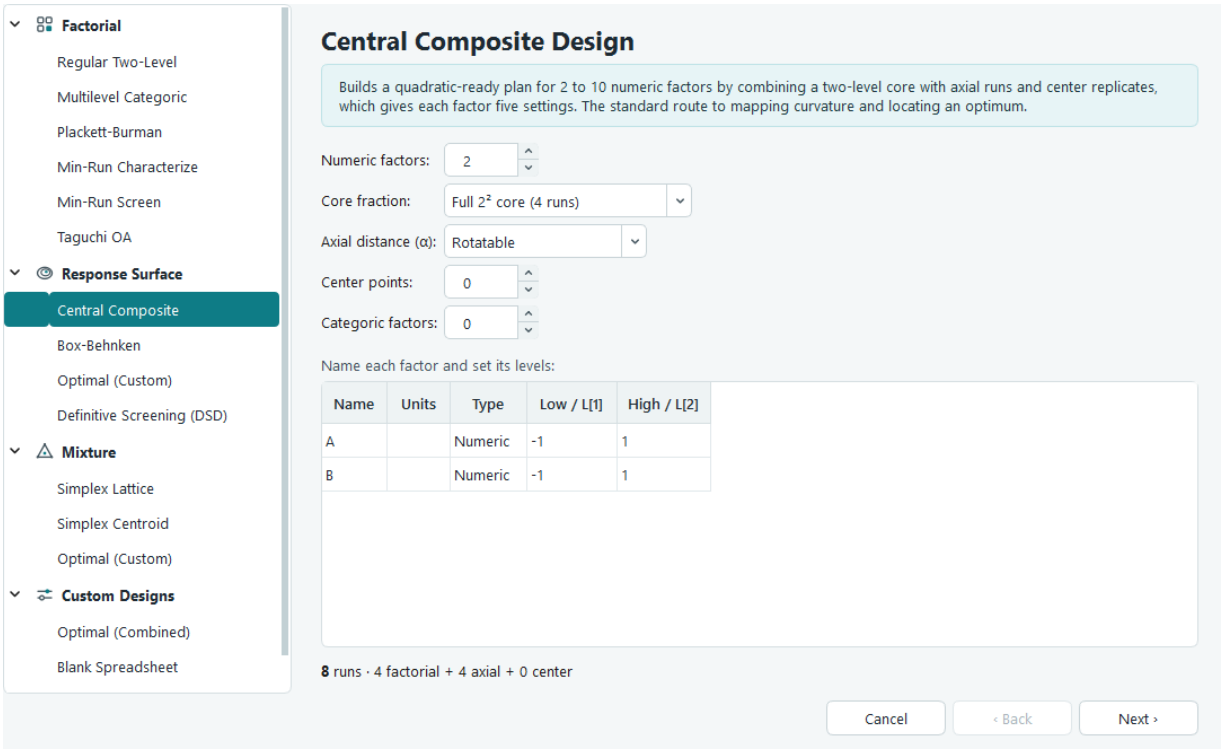
One worked example, followed all the way through. You start with an empty screen and finish with a set of factor settings you can hand to the bench, plus a plan to confirm them. Every step explains why you are doing it, not just where to click.

The problem is a small tablet formulation. The tablet has to fall apart quickly once it is swallowed, so **Disintegration Time** should be short. It also has to survive being packed and handled, so **Hardness** has to stay above a floor. Those two goals pull in opposite directions: the things that make a tablet hard also tend to make it slow to disintegrate. A response surface design is the right tool here because it maps the curved trade-off between the two and lets the optimizer find the compromise.

Two ingredients are in play: **A: HPMC E15**, studied from 2 to 4 % w/v, and **B: Glycerol Ester**, studied from 0.5 to 1.5 % w/v.

> **What you will build.**
> A face centred central composite design on the two factors above, with 5 centre points, for a total of 13 runs, and two responses, Disintegration Time (sec) and Hardness (N).

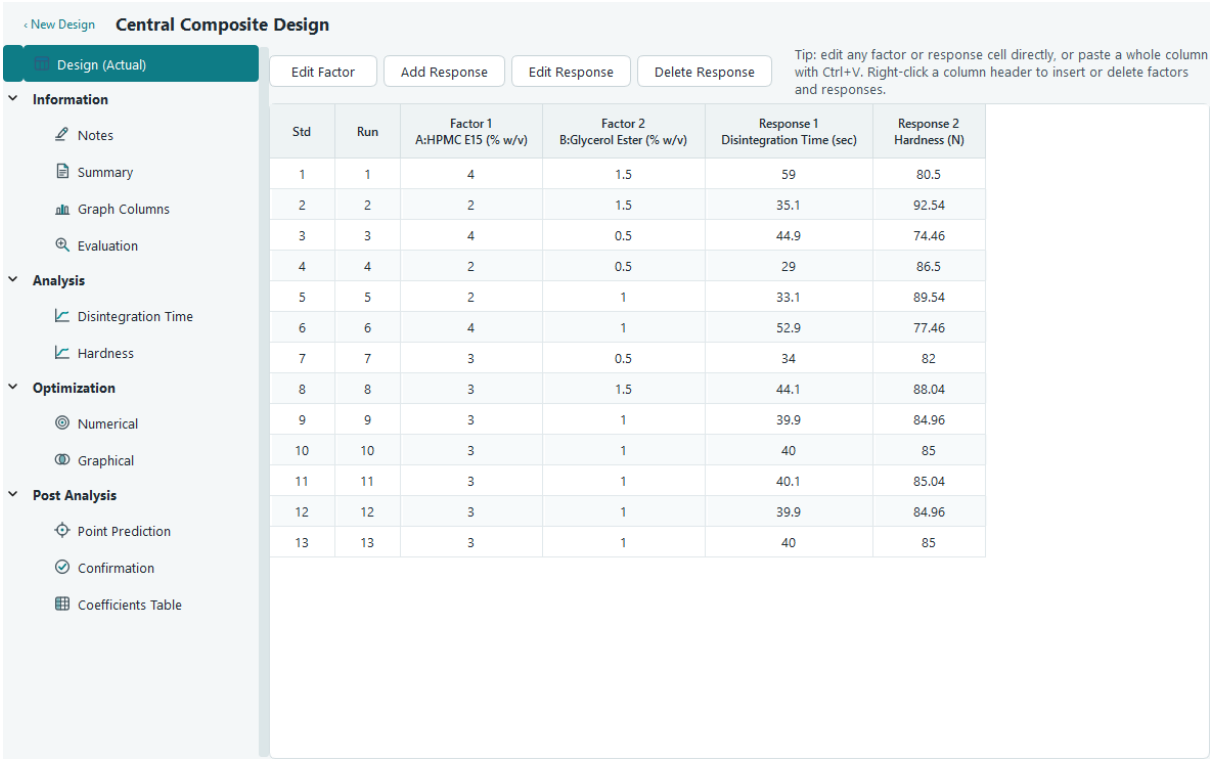
1. **Start a new central composite design.** From the welcome screen choose New Design, then open the Response Surface family and select Central Composite. The panel on the right explains that this design combines a two level core with axial runs and centre replicates. On the setup page set Numeric factors to 2, choose Face-centered ($\alpha = 1$) for the axial distance, and set Center points to 5. Face centred means the axial runs sit on the faces of the square rather than outside it, so each factor only ever takes three settings: its low, its centre and its high. That keeps every run inside the 2 to 4 and 0.5 to 1.5 ranges you can actually make. The live preview reads `13 runs · 4 factorial + 4 axial + 5 center`.



Central Composite setup. The preview line at the bottom updates as you change the options, so you can see the run count before you commit.

2 **Name the factors, set their ranges, add the two responses.** In the factor table give factor A the name HPMC E15 with units % w/v, low 2 and high 4. Give factor B the name Glycerol Ester with units % w/v, low 0.5 and high 1.5. The letters A and B stay fixed as the internal names; the text you type is the label that shows up on every table and plot. Move to the next page and set Number of responses to 2, then name them Disintegration Time with units sec and Hardness with units N. Leave Randomize run order ticked. Press Finish to generate the run sheet.

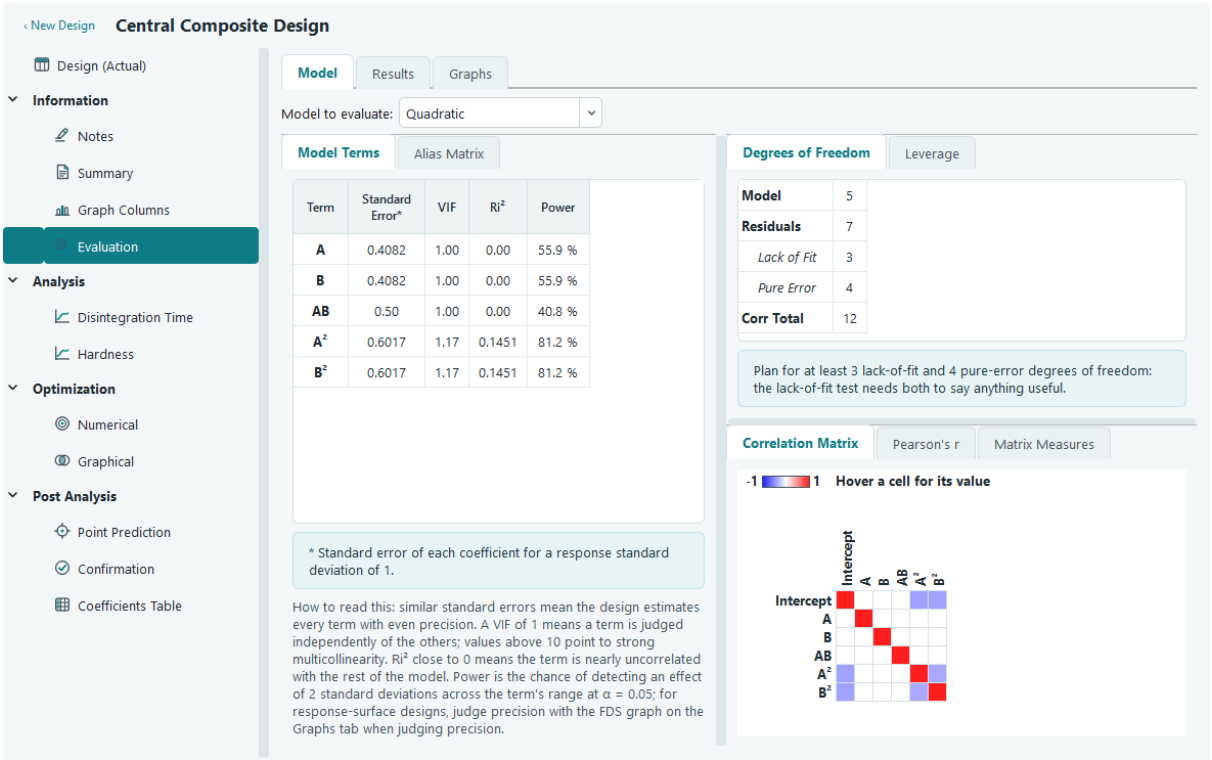
3 **Read the run sheet and note the run order.** The design opens on its run sheet under Design in the navigation pane. Each row is one tablet batch to make and measure. The *Std* column is the textbook order of the design; the *Run* column is the order you should actually work in. The two differ because the run order is randomised. That matters: if anything drifts during the day, such as humidity or a warming press, randomising stops that drift from lining up with a factor and masquerading as a real effect. The response columns on the right are empty and waiting for your measurements.



Std	Run	Factor 1 A:HPMC E15 (% w/v)	Factor 2 B:Glycerol Ester (% w/v)	Response 1 Disintegration Time (sec)	Response 2 Hardness (N)
1	1	4	1.5	59	80.5
2	2	2	1.5	35.1	92.54
3	3	4	0.5	44.9	74.46
4	4	2	0.5	29	86.5
5	5	2	1	33.1	89.54
6	6	4	1	52.9	77.46
7	7	3	0.5	34	82
8	8	3	1.5	44.1	88.04
9	9	3	1	39.9	84.96
10	10	3	1	40	85
11	11	3	1	40.1	85.04
12	12	3	1	39.9	84.96
13	13	3	1	40	85

The run sheet. Column headers read like `A:HPMC E15 (% w/v)` so you can always tell which factor is which.

4 **Check the design before you spend a day making tablets.** Open Information, then Evaluation. This screen inspects the plan on paper, before any data exists. Look at the Model Terms table: the power column tells you the design can estimate each term with even precision, and low variance inflation means the terms are not tangled up with each other. The Leverage and Degrees of Freedom panels confirm the centre points give you enough pure error to test lack of fit later. If something here looked wrong, this is the cheap moment to fix it, by adding runs or centre points, rather than after the experiment.

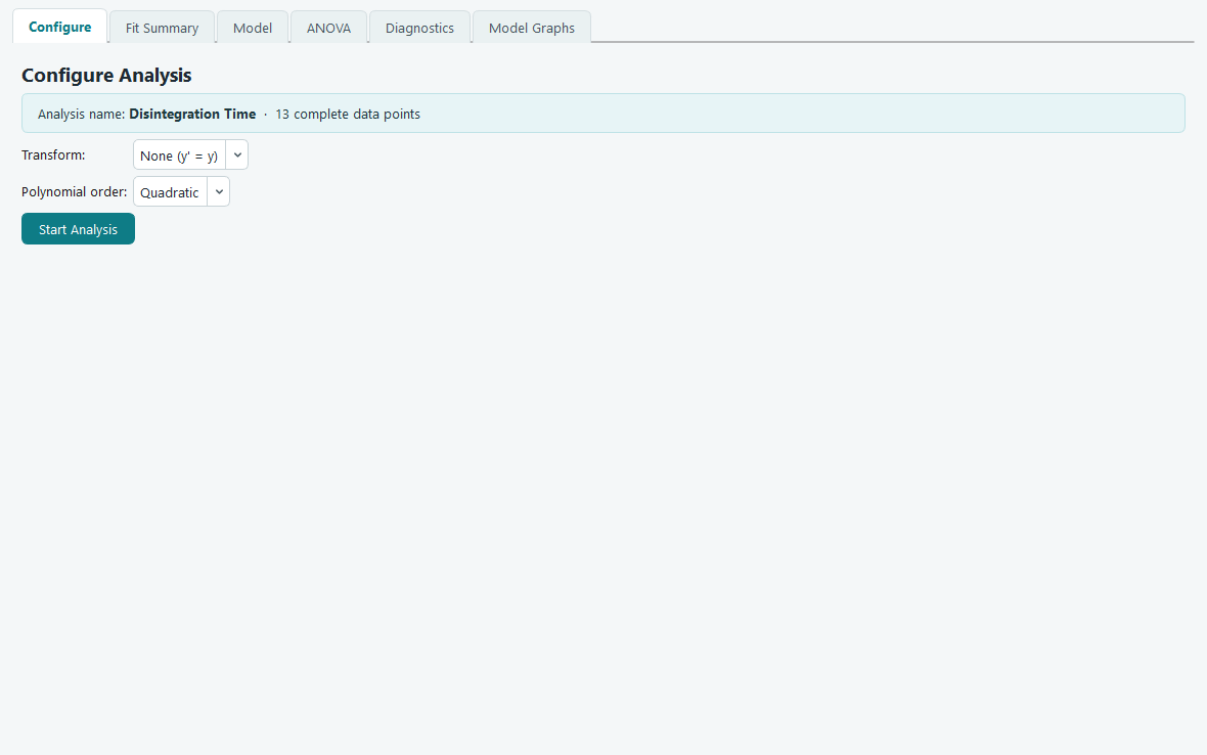


Term	Standard Error*	VIF	Ri²	Power
A	0.4082	1.00	0.00	55.9 %
B	0.4082	1.00	0.00	55.9 %
AB	0.50	1.00	0.00	40.8 %
A²	0.6017	1.17	0.1451	81.2 %
B²	0.6017	1.17	0.1451	81.2 %

Design evaluation. Green through red shading on the correlation grid flags any pair of terms that overlap.

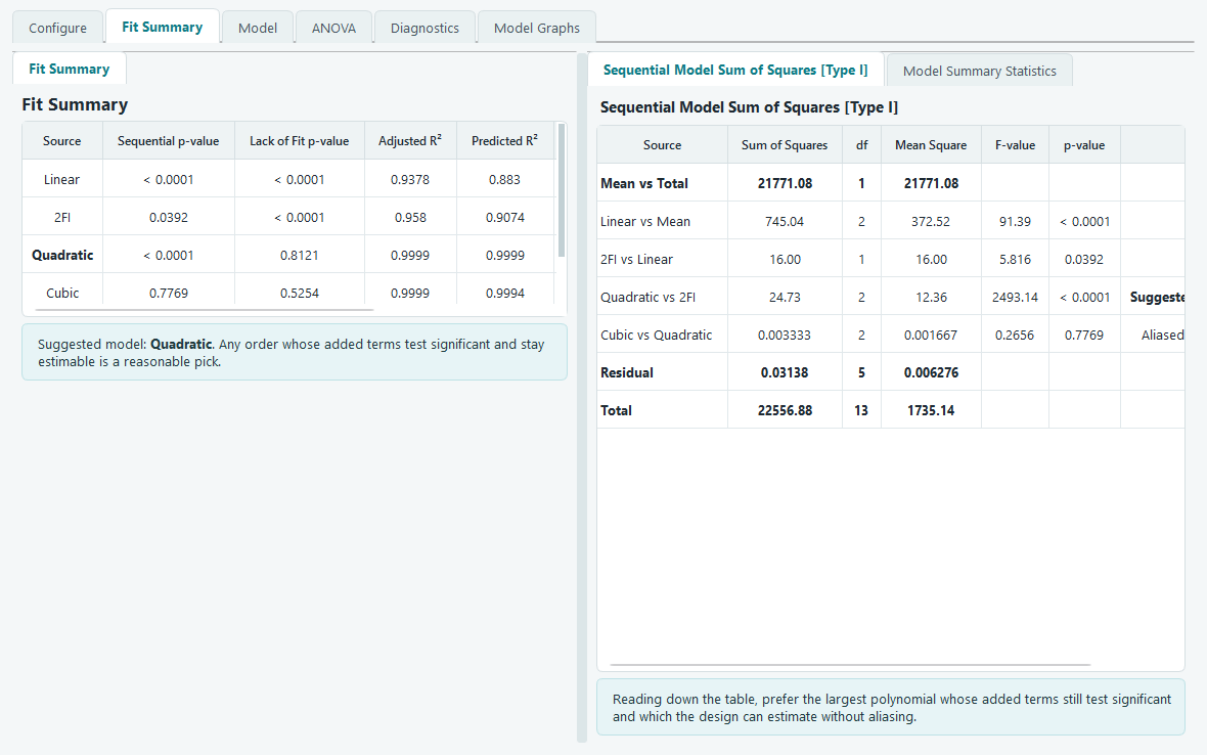
5 **Perform the runs and enter the results.** Make each batch in the Run order shown, measure its disintegration time and hardness, and type the numbers into the matching response cells. You can paste a whole column straight from a spreadsheet with `Ctrl` + `V` , and undo any slip with `Ctrl` + `Z` . The five centre points are deliberately identical recipes; the spread in their results is your honest estimate of pure experimental noise, which the analysis leans on heavily.

6 **Open the first response and start the analysis.** Under Analysis in the navigation pane, click Disintegration Time. You land on the Configure tab. It confirms how many complete data points it found and offers a Transform and a Polynomial order. The defaults are sensible for a first pass, so press Start Analysis. The software fits the model and moves you to Fit Summary.



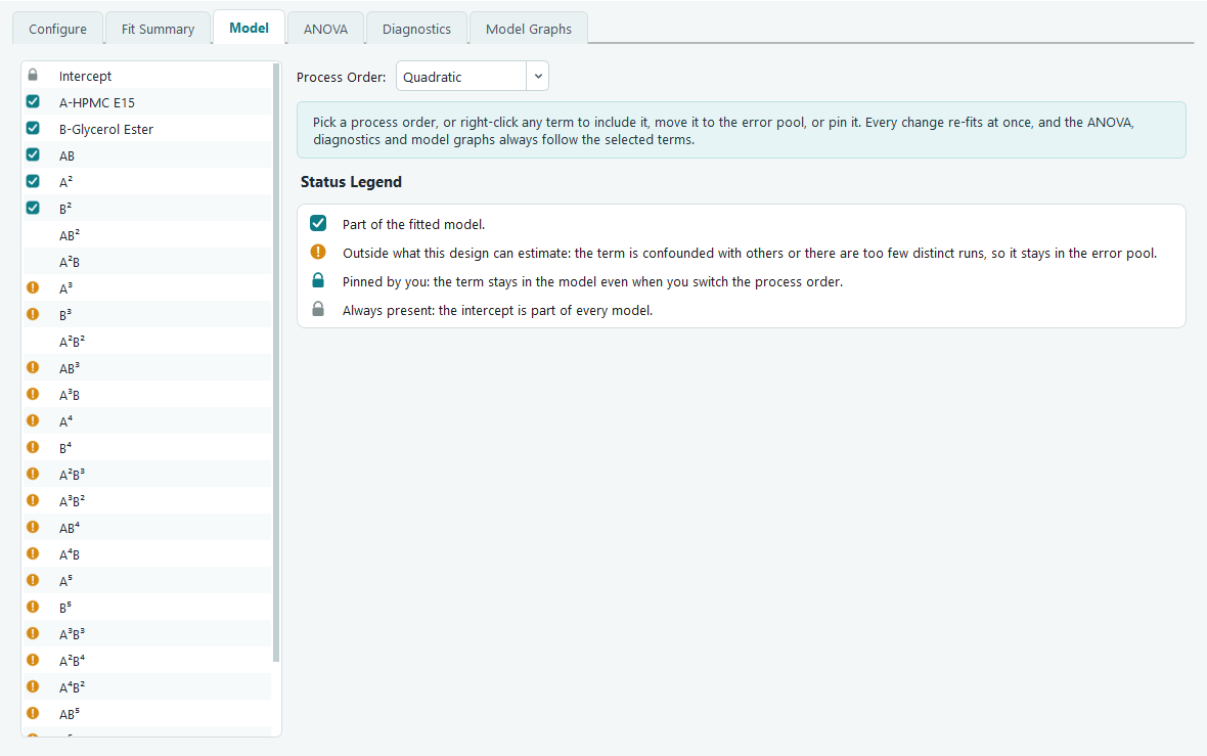
Configure. Changing the order or transform re-arms the Start Analysis button, because the earlier fit is now out of date.

7 **Read the fit summary and take the suggested model.** Fit Summary compares linear, two factor interaction, quadratic and cubic models side by side and marks one as Suggested. For a central composite design the suggestion is usually Quadratic, because that curvature is exactly what the design was built to capture. Two columns guide the choice: the sequential p-value should be small for the order you keep, meaning its extra terms genuinely added something, and the lack of fit p-value should not be small, meaning the model still matches the data. The Adjusted and Predicted R-squared should both be high at the same time. Accepting the suggestion means leaving the order where the software set it.



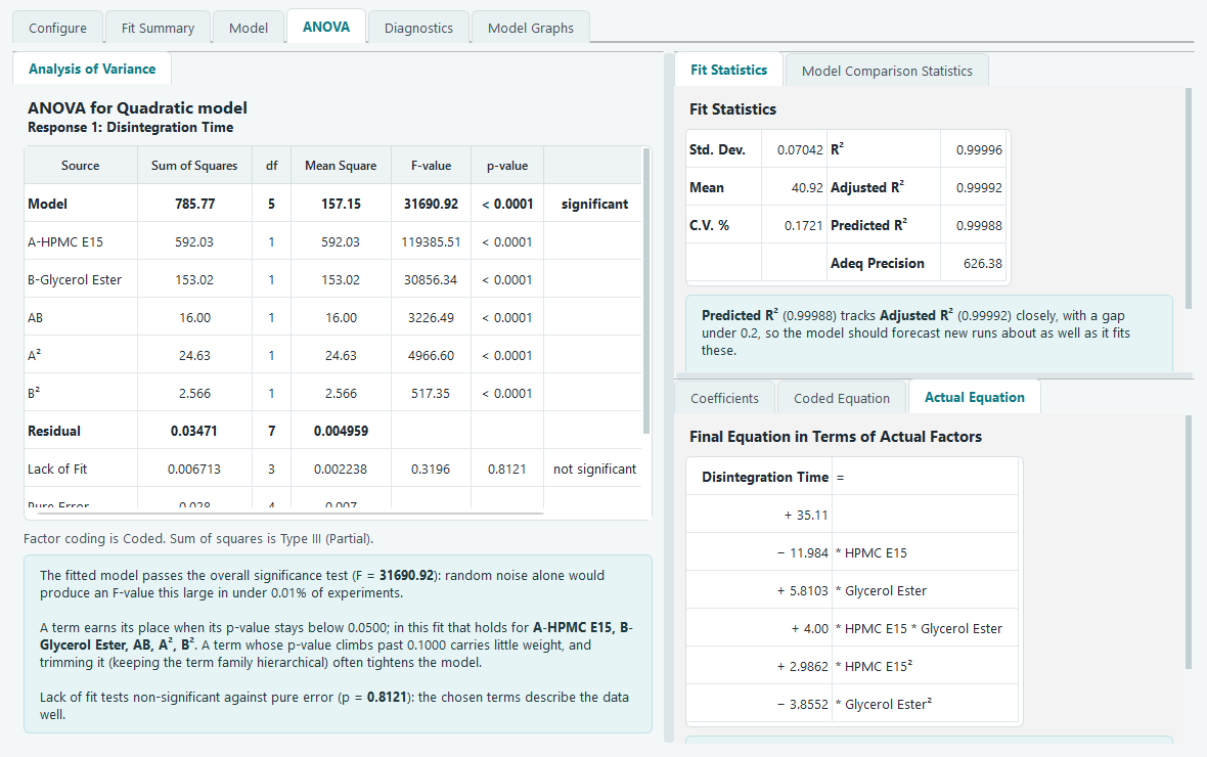
Fit Summary. The suggested row is highlighted, and a short note under the table spells out why in plain language.

8 **Look at the model terms.** The Model tab lists every candidate term with a status mark. A teal check means the term is in the fitted model; an amber dot warns that the design cannot estimate that term; the intercept carries a lock because it is always present. For a two factor quadratic you should see the intercept, the main effects A and B, the interaction AB, and the squared terms for both factors. You can right click any term to move it in or out, and the whole analysis re-fits at once, but for now the suggested set is a good starting point.



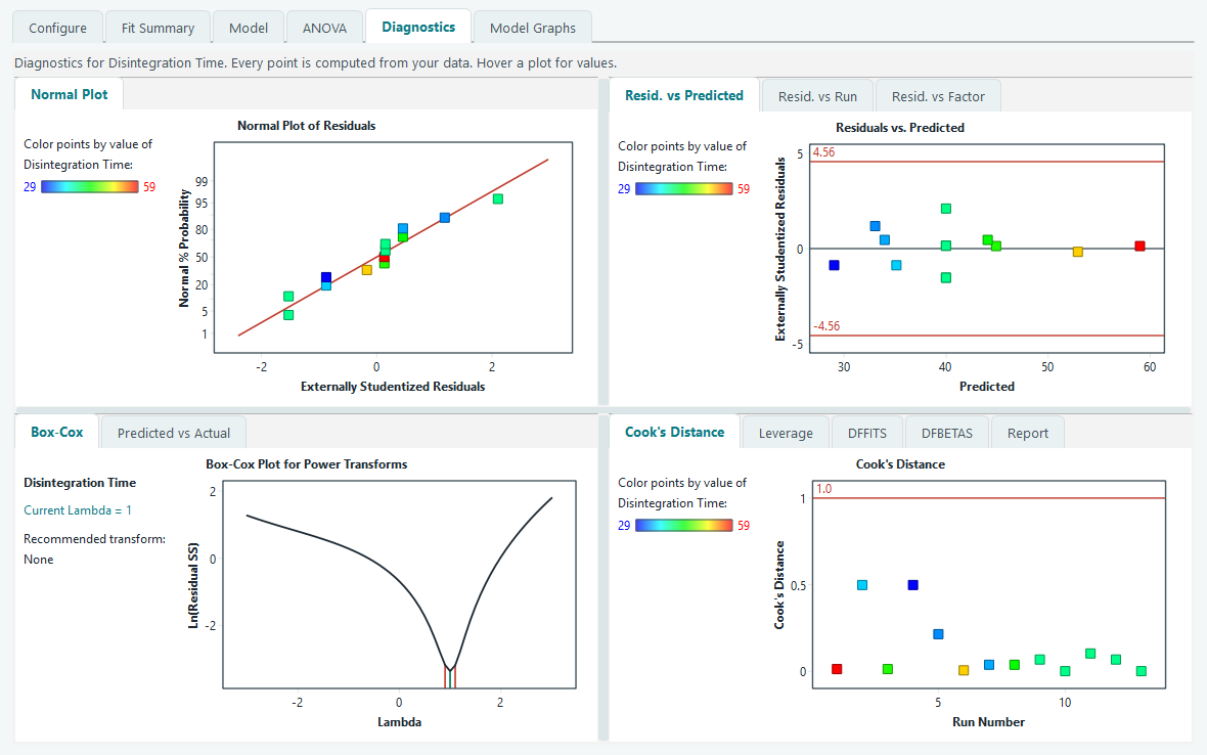
Model terms. Changes here flow straight through to the ANOVA, diagnostics and graphs.

9 **Read the ANOVA and the equation.** The ANOVA tab tests the model as a whole and each term on its own. Read the top Model row first: if its p-value is below 0.05 the surface you fitted is unlikely to be an accident of noise. Then scan the individual rows to see which of A, B, AB and the squared terms carry their weight, again at p below 0.05. The Lack of Fit row is the reassuring one: you want it not significant, which says the model misses nothing the pure centre point error can detect. Every table on this tab comes with a written interpretation underneath, computed from your own numbers. The Actual Equation writes the fitted model out in real units, so you can predict a disintegration time by plugging in an HPMC and a glycerol ester level directly.



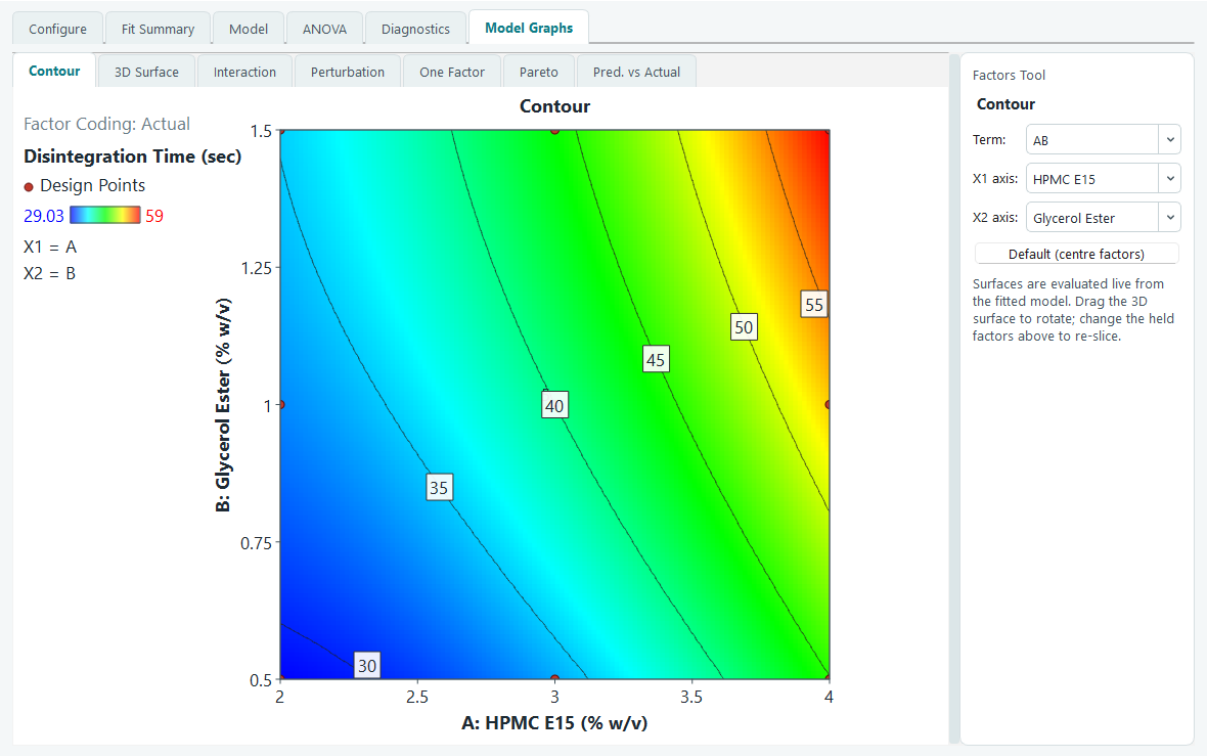
The equation in actual factors. Judge which factor matters most from the coded equation instead, since unit scaling distorts these coefficients.

10 **Check the diagnostics.** Statistics only mean something if the model's assumptions hold, and the Diagnostics tab is where you check them. On the normal plot the residuals should fall close to a straight line. On residuals versus predicted they should scatter in a shapeless band, with no funnel that would hint the spread grows with the response. On predicted versus actual the points should hug the diagonal. The influence plots and the Report table flag any single run that is pulling the fit around more than its share. If everything looks calm, the model is trustworthy.

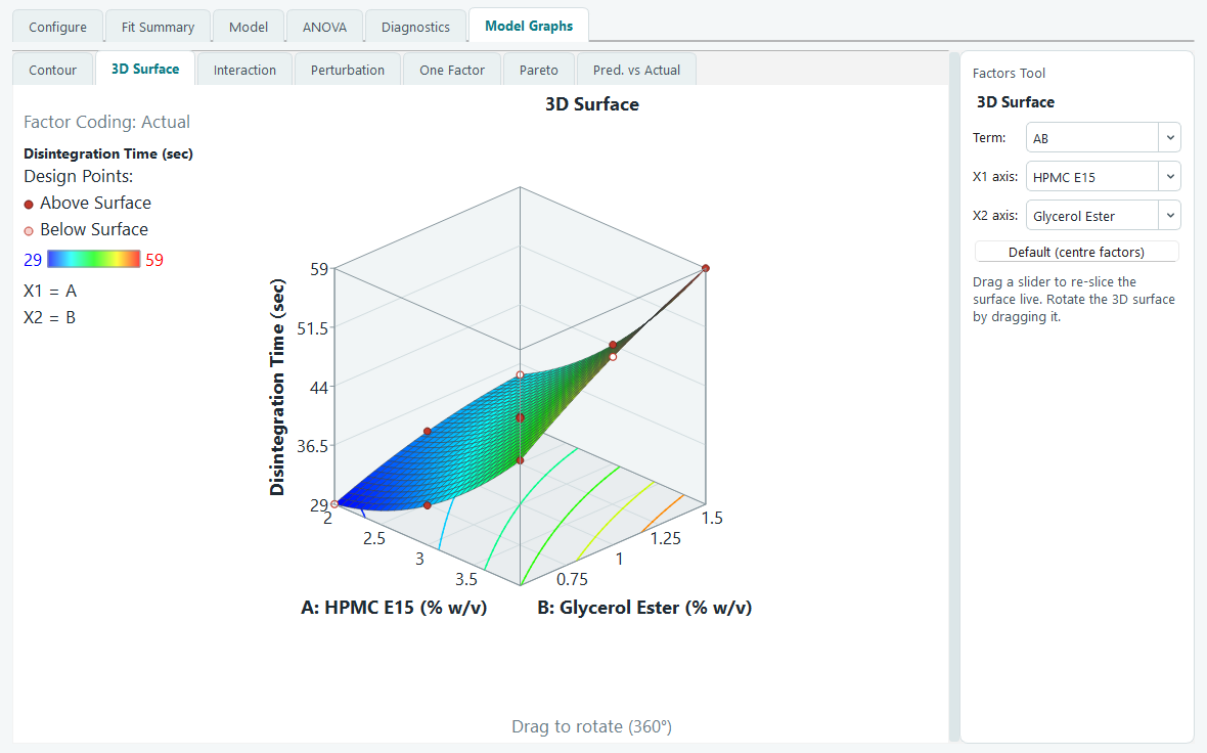


Diagnostics. A pattern here is a signal to try a transform or reconsider the terms, not to ignore it.

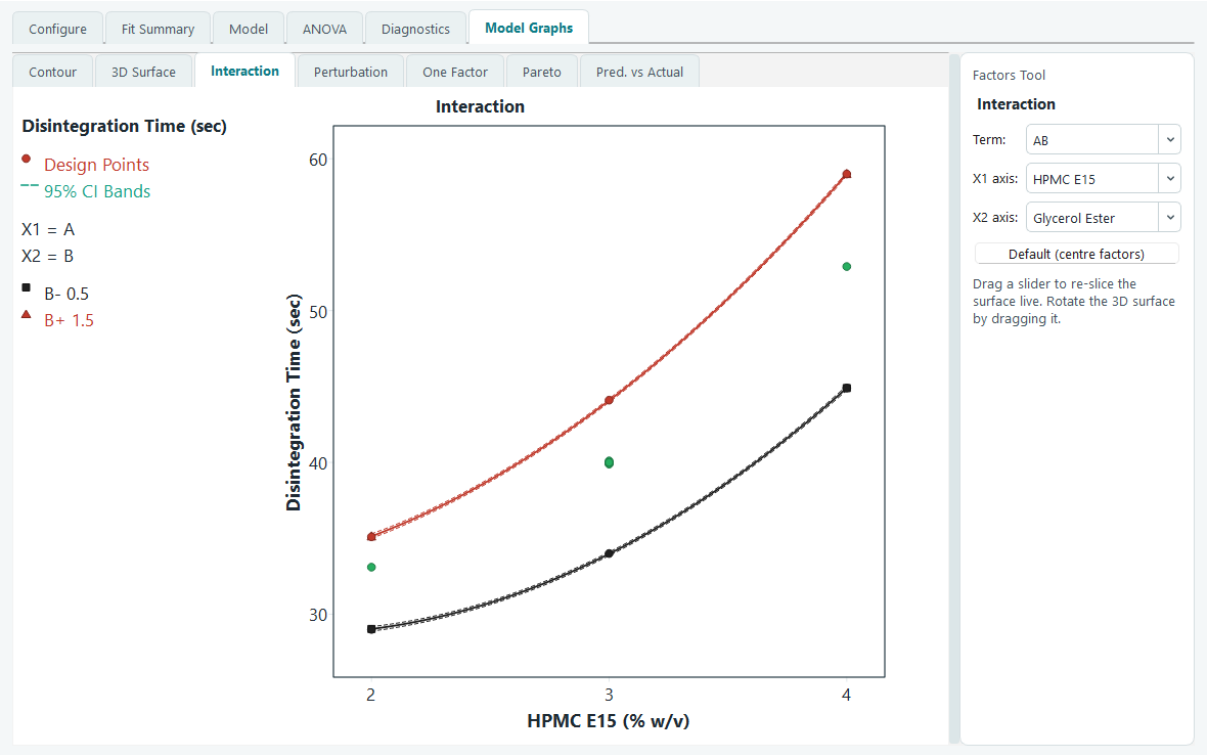
11 **Look at the model graphs.** Now see the surface. The Contour tab draws the map you will put in a report: lines of equal disintegration time across the HPMC and glycerol ester plane. The 3D Surface makes the shape and its curvature obvious at a glance. The Interaction plot answers a specific question: if the two lines are far from parallel, the effect of one factor depends on the level of the other, which is the AB term earning its place. The Factors Tool on the right lets you slide any factor you are not plotting and watch the surface re-slice live.



The contour map of Disintegration Time.



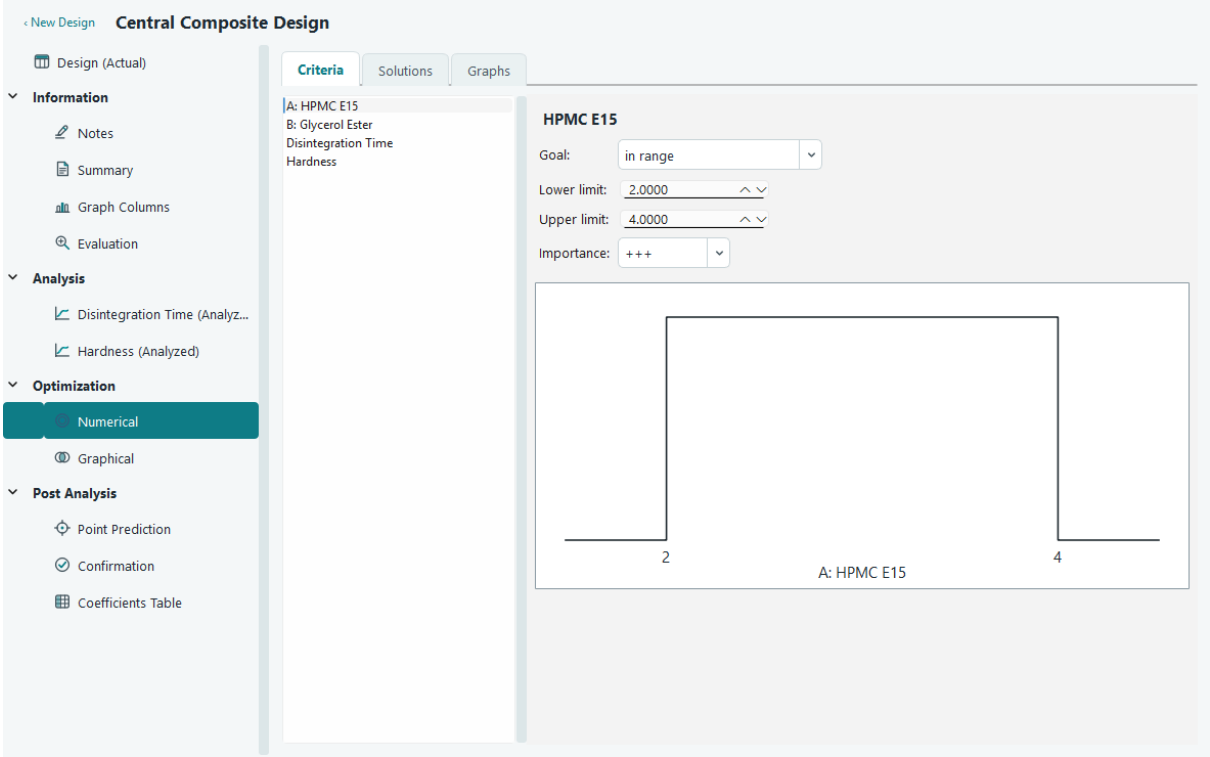
The same fit as a 3D surface, with the design points shown above and below it.



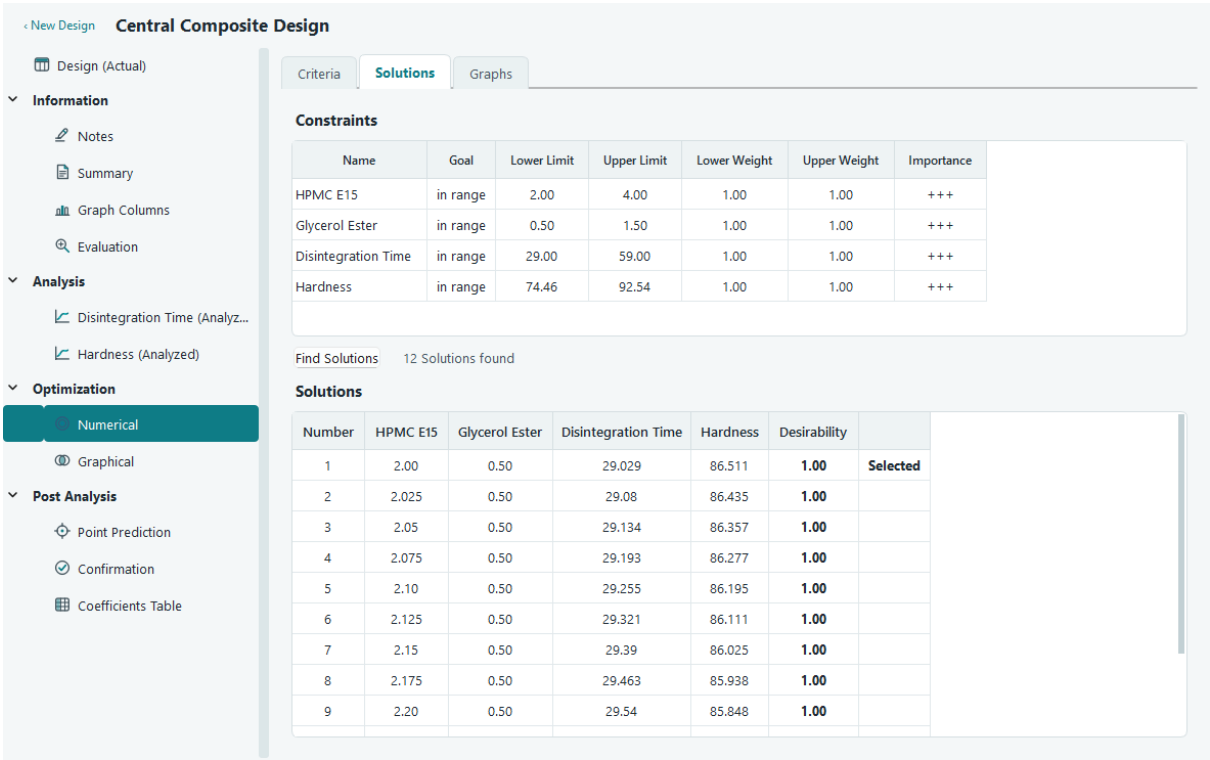
The interaction plot. Non parallel lines mean the two ingredients are not acting independently.

12 **Repeat for the second response.** Click Hardness under Analysis and run through the same short loop: Configure, Start Analysis, take the suggested model, glance at the ANOVA, confirm the diagnostics behave. You now have two fitted models over the same design space, one for speed and one for strength. The optimizer needs both before it can look for a compromise.

13 **Set goals and run the numerical optimizer.** Open Optimization, then Numerical. The Criteria list holds every factor and every analysed response. Leave the two factors set to in range, so the search is free to pick any recipe inside the studied box. Set Disintegration Time to minimize. For Hardness, keep it in range but raise the lower limit to the softest tablet you can accept, so any recipe below that floor is rejected outright. Importance and the desirability ramp let you say how hard each goal is pulling. Switch to Solutions and press Find Solutions. The optimizer combines the goals into a single desirability score between 0 and 1 and lists the settings that score highest, best first, with the top row marked Selected.

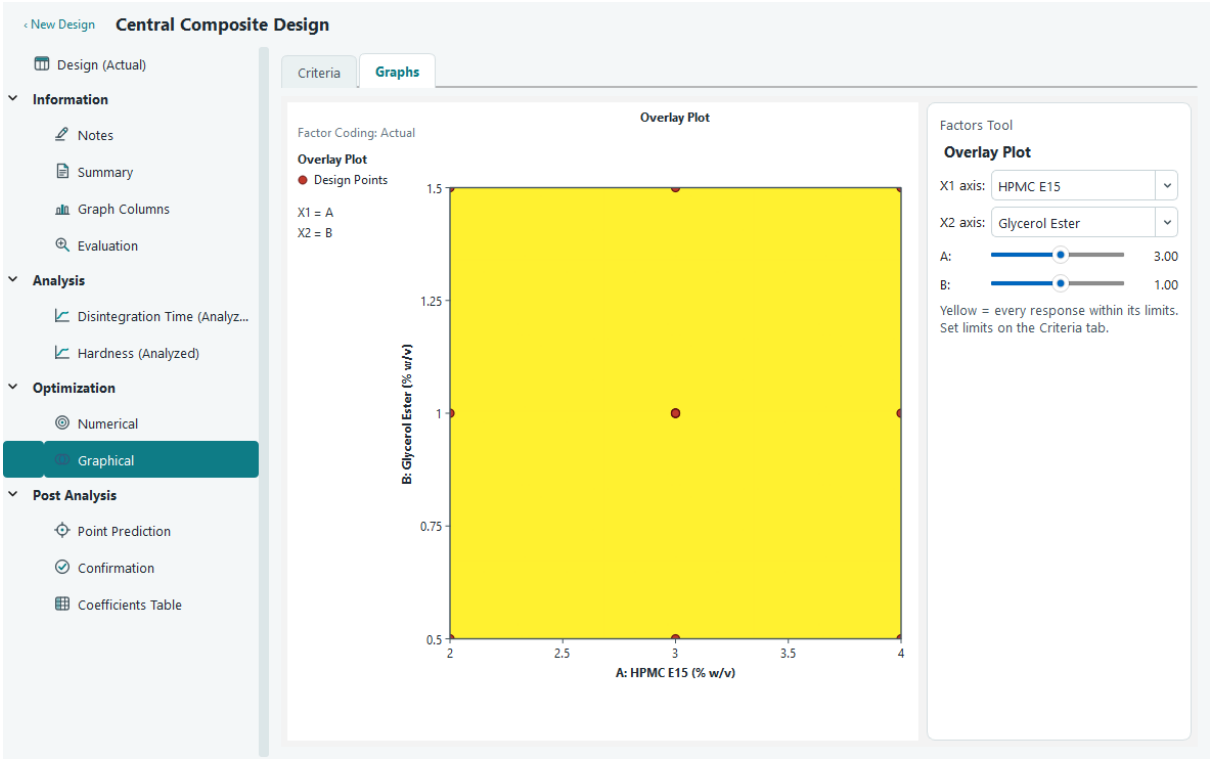


Goals for each response. The ramp on the right shows the desirability shape you are asking for.



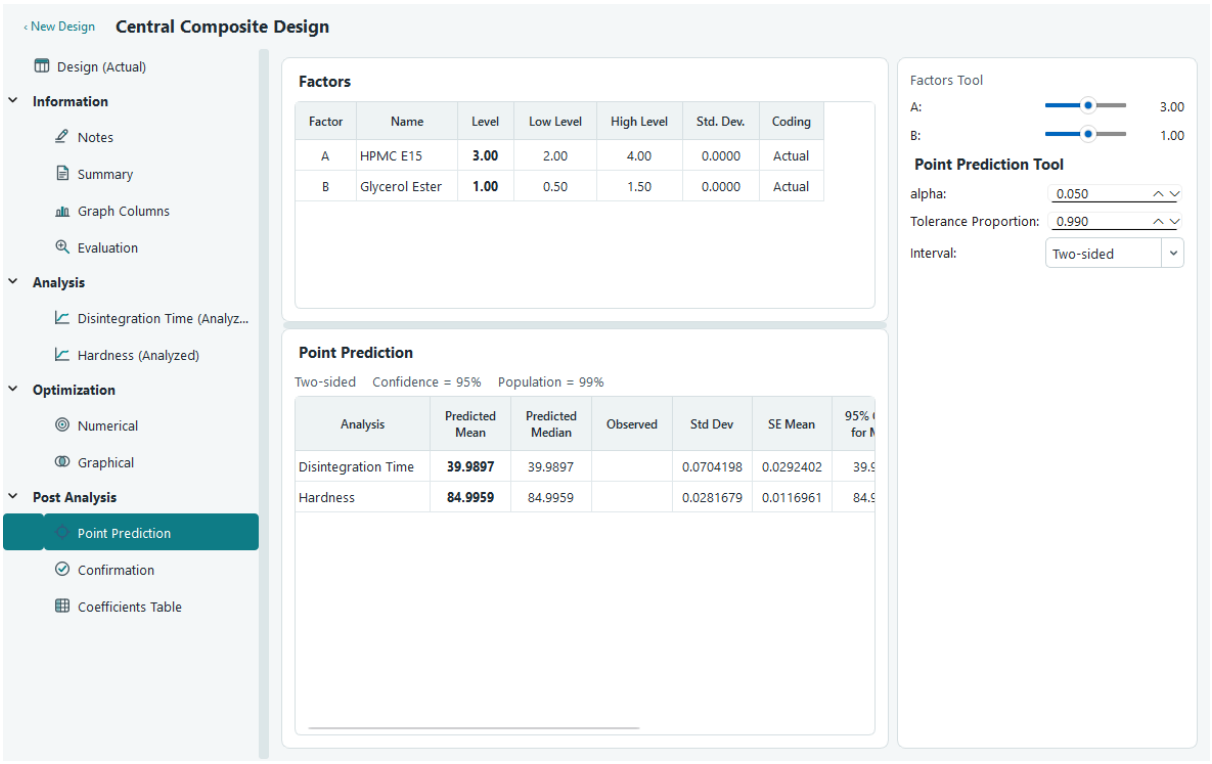
Solutions, best first. Each row is a recipe you could actually make, with its predicted responses and overall desirability.

14 **Look at the overlay plot to see how much room you have.** A single best point can be fragile. Open Graphical optimization, set a lower and upper limit on each response to match your specification, and read the overlay plot. The yellow region is where every response stays inside its limits at once. A wide yellow patch means you have comfortable room to move if a batch drifts; a thin sliver means the recipe is delicate and worth extra control. Pick a working point near the middle of the yellow, not on its edge.

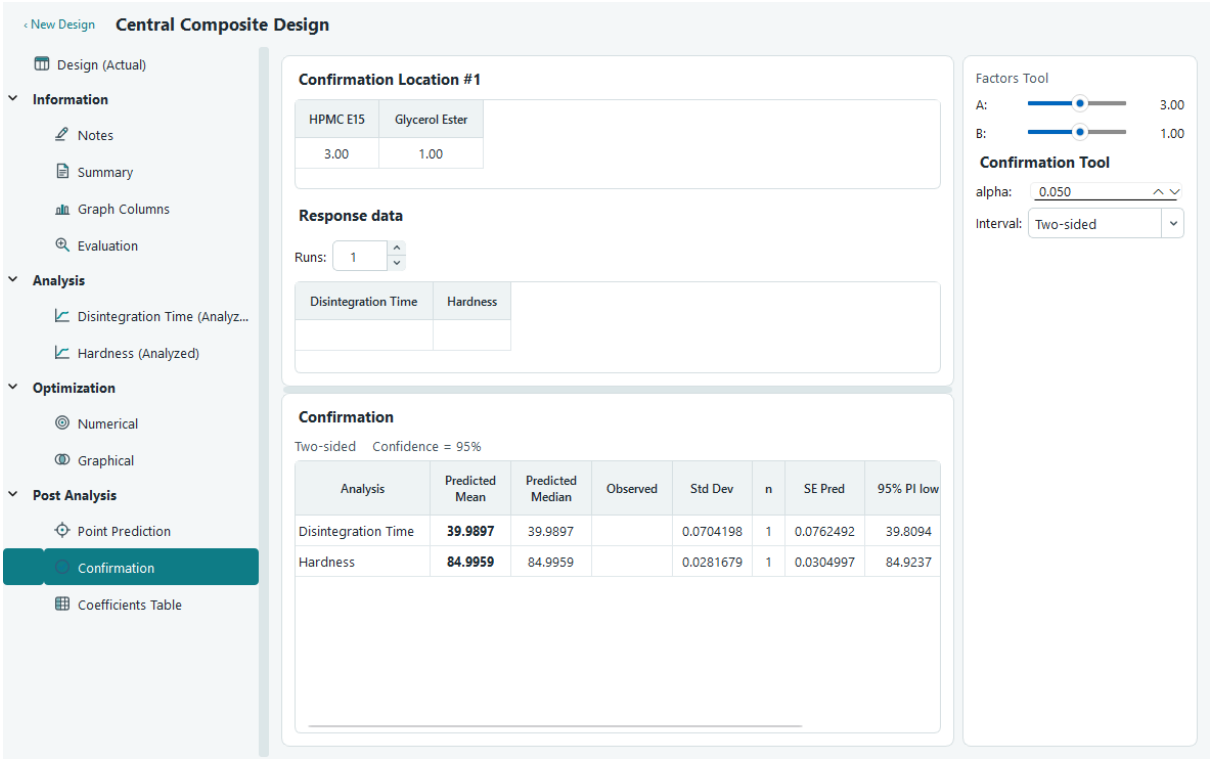


The overlay plot. Yellow is the sweet spot where speed and strength are both acceptable.

15 **Predict at the chosen settings, then plan confirmation runs.** Under Post Analysis, open Point Prediction. Slide the factors to your chosen HPMC and glycerol ester levels and read the predicted mean for each response, along with its confidence interval for the mean and a tolerance interval for the population. The interval, not just the point, is what you promise. Finally open Confirmation, make a few tablets at exactly those settings, and type their measured results in. The screen compares your batch mean against the model's prediction interval. If it lands inside, the model held up and you are done. If it falls outside, the surface needs another look before you trust it in production.



Point prediction at the chosen recipe, with intervals around each mean.



Confirmation. Enter real runs and see whether their mean sits inside the predicted interval.

What you learned

- A face centred central composite design maps curvature on two factors in only 13 runs, with 5 centre points carrying the pure error estimate.
- Randomised run order protects you from drift during the experiment.

- Fit Summary suggests a model, and the ANOVA, diagnostics and graphs tell you whether to trust it.
- The numerical optimizer turns competing goals into one desirability score, and the overlay plot shows how much room the winning settings leave you.
- Point prediction gives you an interval to promise, and confirmation runs prove the model on the bench.

Where to go deeper

- Fit summary, model and ANOVA for what every statistic on those tables means.
- Diagnostics for reading each residual and influence plot.
- Numerical optimization and graphical optimization for goals, weights and limits in full.
- Prediction and confirmation for the intervals behind the last step.

Chapter 19

Screening with a factorial

Early in a project you often have a long list of factors and no idea which ones actually move the response. Screening is the cheap first pass that sorts them out. You test all of them together in a small number of runs, then keep only the few that matter. This tutorial walks that pass from end to end with a two-level factorial.

Why screen first

Studying every factor in full detail is expensive. If you have six candidate factors, mapping each one carefully could cost dozens of runs before you even know whether half of them do anything. A screen turns the question around. Instead of asking "what is the exact shape of each effect", it asks "which factors have any effect worth chasing". Once you know the vital few, you can spend your remaining runs on those and drop the rest.

A two-level factorial is built for this. Every factor is set to just a low and a high value, so each factor costs very little, and you can pack many of them into one compact plan. The price of that economy is *resolution*: in a small plan some effects get tangled together, and part of the skill is choosing a plan where the tangles do not hide what you care about.

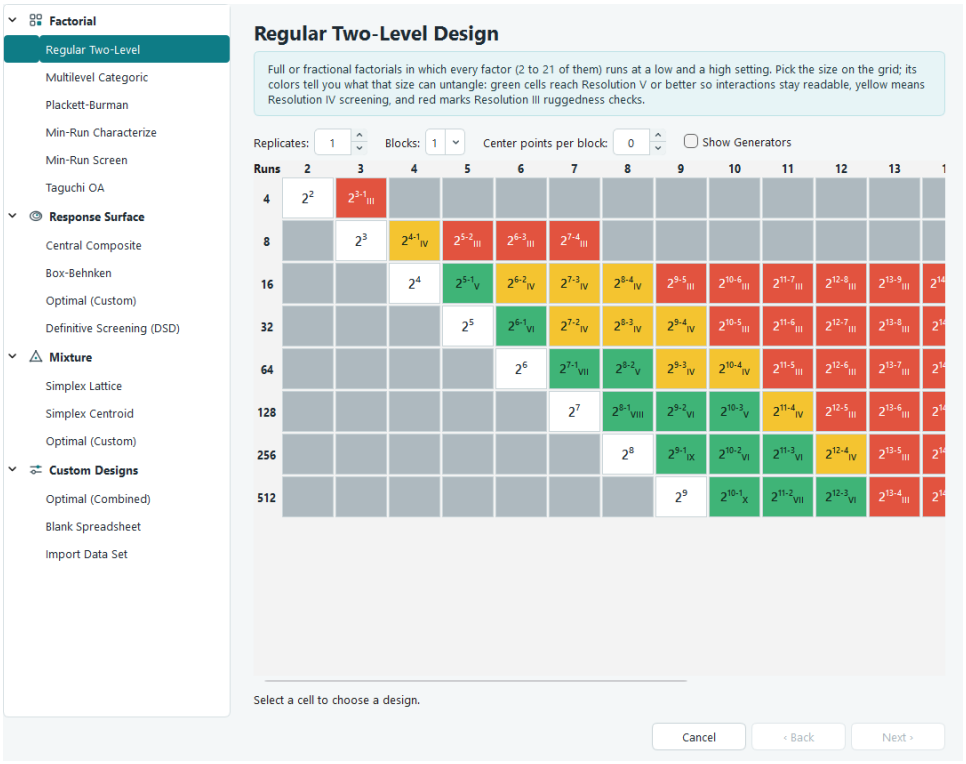
The example

Say you are developing a film coating for a tablet and six things might affect how fast the tablet disintegrates: the HPMC E15 level, the glycerol ester (plasticiser) level, the coating temperature, the spray rate, the pan speed and the curing time. You measure one response, Disintegration Time in seconds. The goal here is not to optimise anything yet. It is to find out which two or three of these six factors are worth a proper study.

> **Shortcut.**
> If you would rather answer a few questions than choose a design yourself, open the Design Wizard from the Design Tools menu and select the *Screening* goal. It recommends a screening design for your factor count and run budget and then builds it for you.

Step by step

1. From the welcome screen choose **New Design**, then pick **Regular Two-Level** under the Factorial family. The panel on the right explains what the design is for and how to read the grid you are about to see.
2. You now choose the size of the plan on a grid. Each cell is one possible design: how many factors across the top, how many runs down the side. The cell colours are the important part. Green cells reach Resolution V or better, so main effects and two-factor interactions all stay readable. Amber cells are Resolution IV, which keeps main effects clean but lets two-factor interactions overlap each other. Red cells are Resolution III, the cheapest option, where main effects themselves can be confounded with interactions.



The two-level grid. Pick a cell, then read the line underneath: it names the design, the run count and the resolution in the same colour as the cell.

For six factors, the 16-run cell is a good screening choice. The software marks it a quarter fraction at Resolution IV (amber): every main effect comes out clean, which is exactly what a screen needs. Tick **Show Generators** if you want to see the defining relation that spells out which effects share a column. You can also set replicates, blocks and centre points here, but for a first screen the defaults are fine.

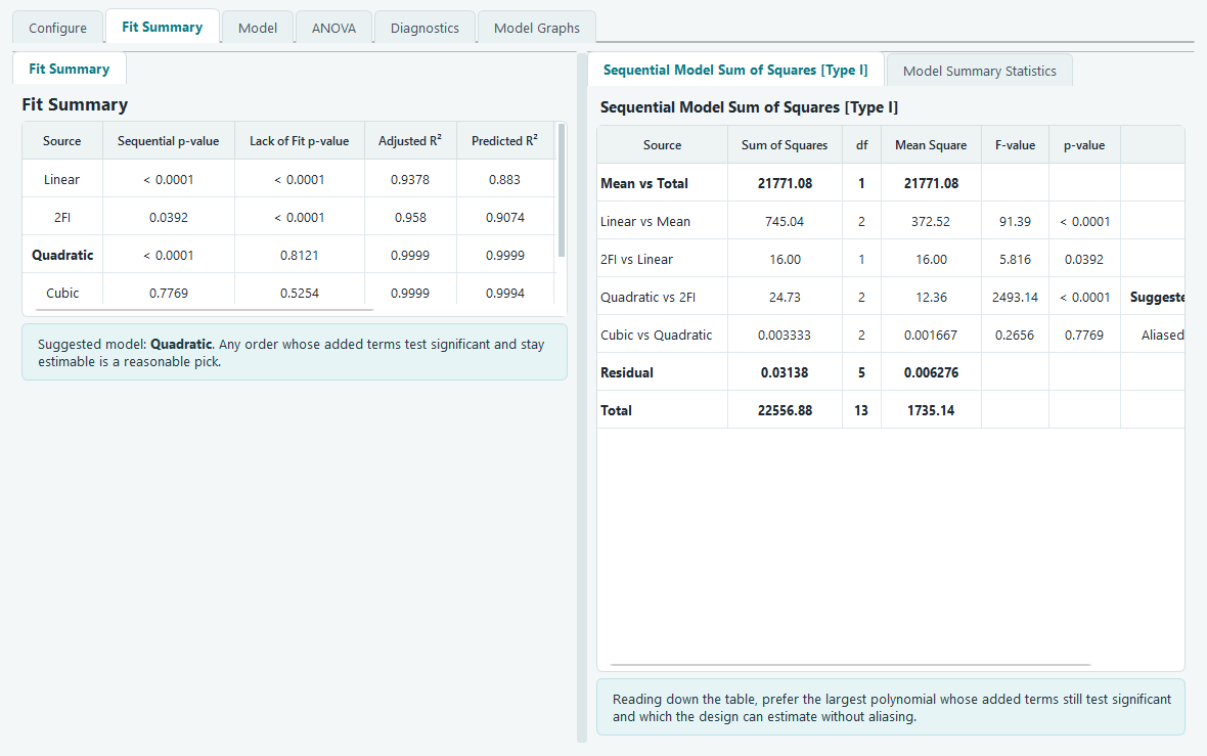
> **Resolution is a trade.**
> A greener cell answers more questions but costs more runs. Screening usually lives in the amber band on purpose: you accept that interactions are blurred together, because at this stage you only want to know which factors matter, not how they combine.

3. Press Next and name your factors. Give each one a label, its units, and the low and high settings you will actually run. The letter (A, B, and so on) stays as the factor's short name, and the label you type is what appears everywhere else, so a run sheet column reads like `A:HPMC E15 (% w/v)`.
4. On the last page, define your response. Here that is Disintegration Time in seconds. Leave **Randomize run order** ticked, then press Finish to build the run sheet.

> **Why the order is scrambled.**
> The *Run* column is the order in which to perform the experiments, and it is deliberately randomised. If anything drifts while you work, a warming room or an ageing reagent, randomising stops that drift from lining up with any one factor and being mistaken for its effect.

5. Perform the runs in the order shown and type each measured Disintegration Time into the response column. You can paste a whole column from a spreadsheet with `Ctrl` + `V`, and undo any slip with `Ctrl` + `Z`.
6. Open the response under **Analysis** in the navigation tree. The tabs run left to right in the order you use them: Configure, Fit Summary, Model, ANOVA, Diagnostics, Model Graphs. Start on Fit Summary. It

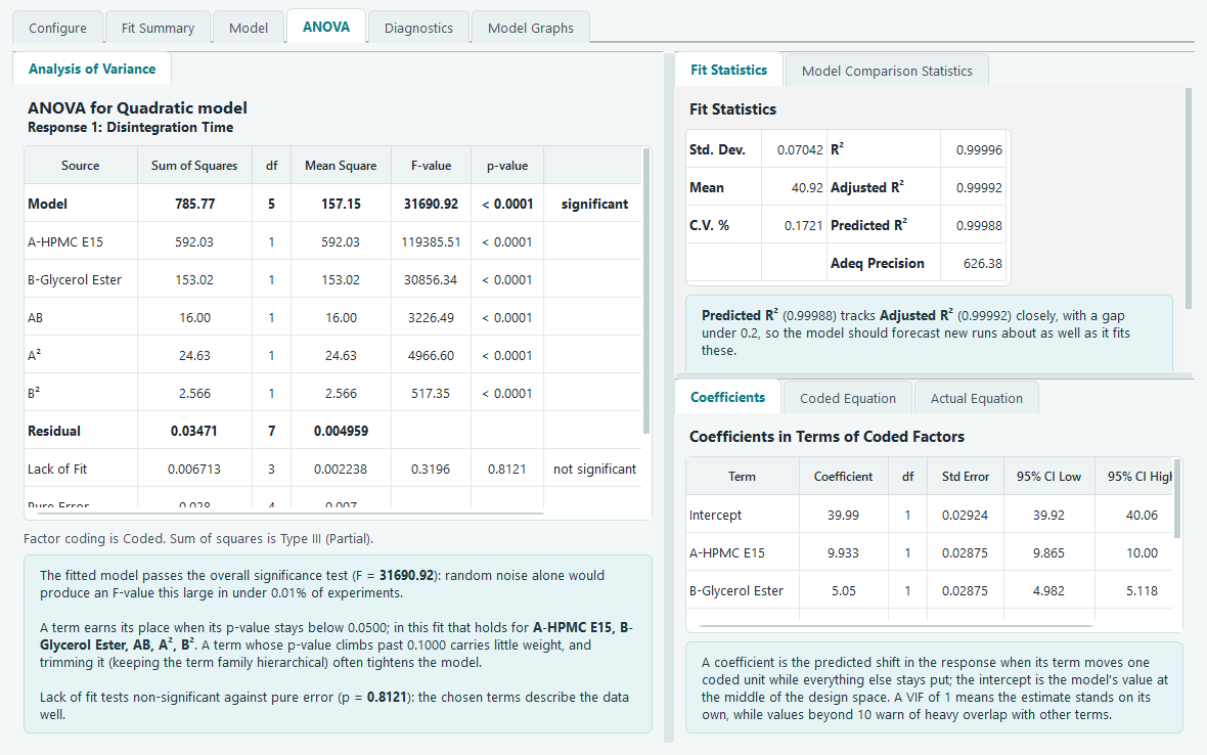
compares the models the design can support and flags the one the data backs, marking anything the plan cannot separate as aliased. For a screen the answer is usually a main-effects model, sometimes a two-factor interaction model if a couple of interactions stand out.



Fit Summary suggests a model and marks the terms the design can and cannot estimate cleanly.

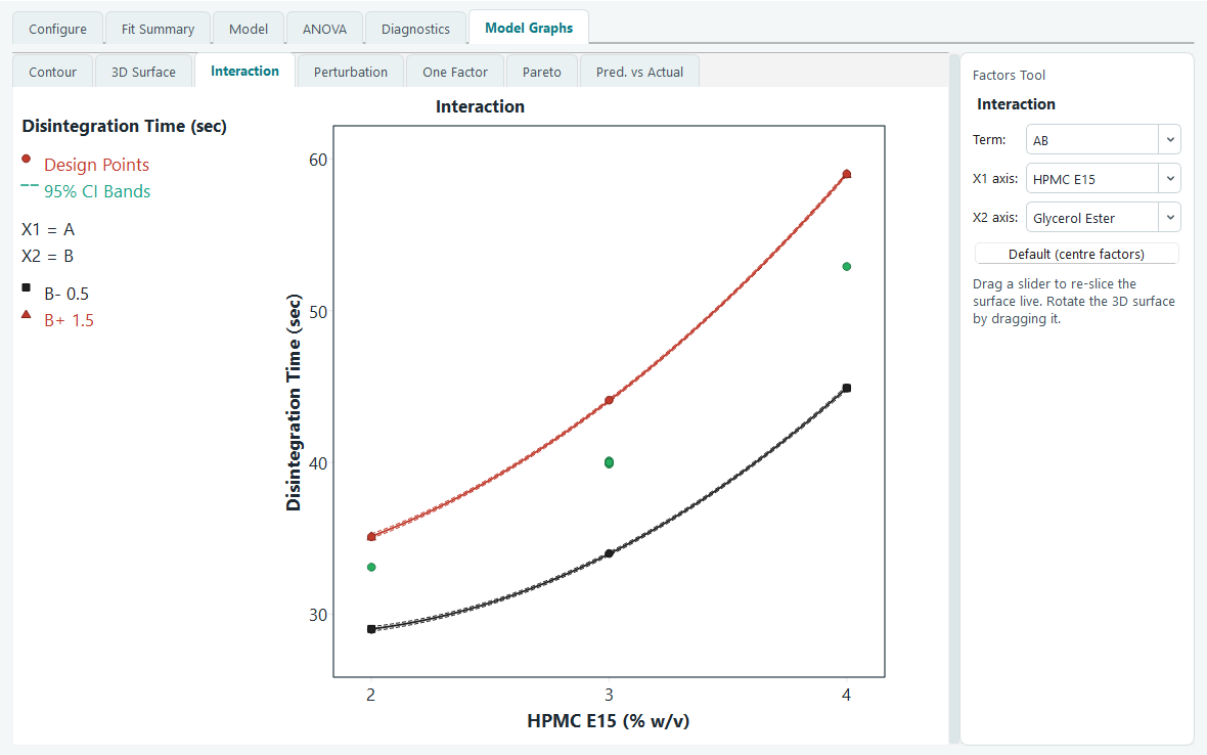
> **About these screenshots.**
> The analysis pictures on this page come from the response surface example used elsewhere in this guide, which studies HPMC E15 and glycerol ester in detail. The analysis screens look and behave the same whatever design produced the data, so read them for the layout, not for the exact numbers.

7 Move to the **ANOVA** tab. This is where a screen pays off. The table lists every term with its F-value and p-value, and the plain-language note underneath tells you which terms carry weight. A term earns its place when its p-value stays below 0.0500. Those significant main effects are your vital few. Any factor whose p-value climbs past 0.1000 is probably doing nothing, and you can drop it.



ANOVA, with the significance verdict written out below the table. The terms that stay under 0.0500 are the factors worth keeping.

8 If two of the surviving factors both matter, check whether they interact. Open **Model Graphs** and choose the interaction plot. Two lines that stay parallel mean the factors act independently. Lines that cross or fan apart mean the effect of one factor depends on the setting of the other, which is worth knowing before you plan the next stage.



An interaction plot. Non-parallel lines say the two factors do not act independently.

Reading the result

At the end of the screen you have a short list. In the coating example, suppose the ANOVA shows that HPMC E15 and glycerol ester dominate the disintegration time while temperature, spray rate, pan speed and curing time all test non-significant. That is the whole point of screening: four factors have been cleared away, and you can now focus every remaining run on the two that count.

Do not over-read a screen.
A Resolution IV plan blurs two-factor interactions together, so an amber design cannot tell you exactly which interaction is real, only that main effects are present. Treat significant effects as leads to follow, not as final conclusions. Confirmation and a richer design come next.

What to do next

Once you know the vital few, you change goal. Screening was about which factors; the next stage is about how they behave.

- If a couple of factors clearly matter and you now want the shape of the response and an optimum, move to a response surface design on just those factors. That is exactly the path the main tutorial takes: it studies HPMC E15 (2 to 4 % w/v) and glycerol ester (0.5 to 1.5 % w/v) with a face-centred central composite design. See Response surface, start to finish .
- If you have three to five survivors and want to characterise how they interact before optimising, a full two-level factorial or a Min-Run Characterize design keeps every interaction readable. See Factorial designs .
- Before you spend money on the next set of runs, it is worth checking the plan on the evaluation screen, and confirming a promising setting with a confirmation run .

Chapter 20

A mixture experiment

Some experiments are about a recipe, not a machine. You have three ingredients that must add up to the whole blend, and you want to know how their proportions change the result. This walk through builds a three component blend, enters the results, fits the mixture model, and reads the surface, using a small tablet formulation as the example.

When the ingredients are the experiment

Suppose you are formulating a tablet filler from three excipients: lactose, microcrystalline cellulose and maize starch. Together they make up 100% of the blend, and you want to know how their proportions affect the disintegration time. This is a mixture problem, and it behaves differently from a factorial or a response surface.

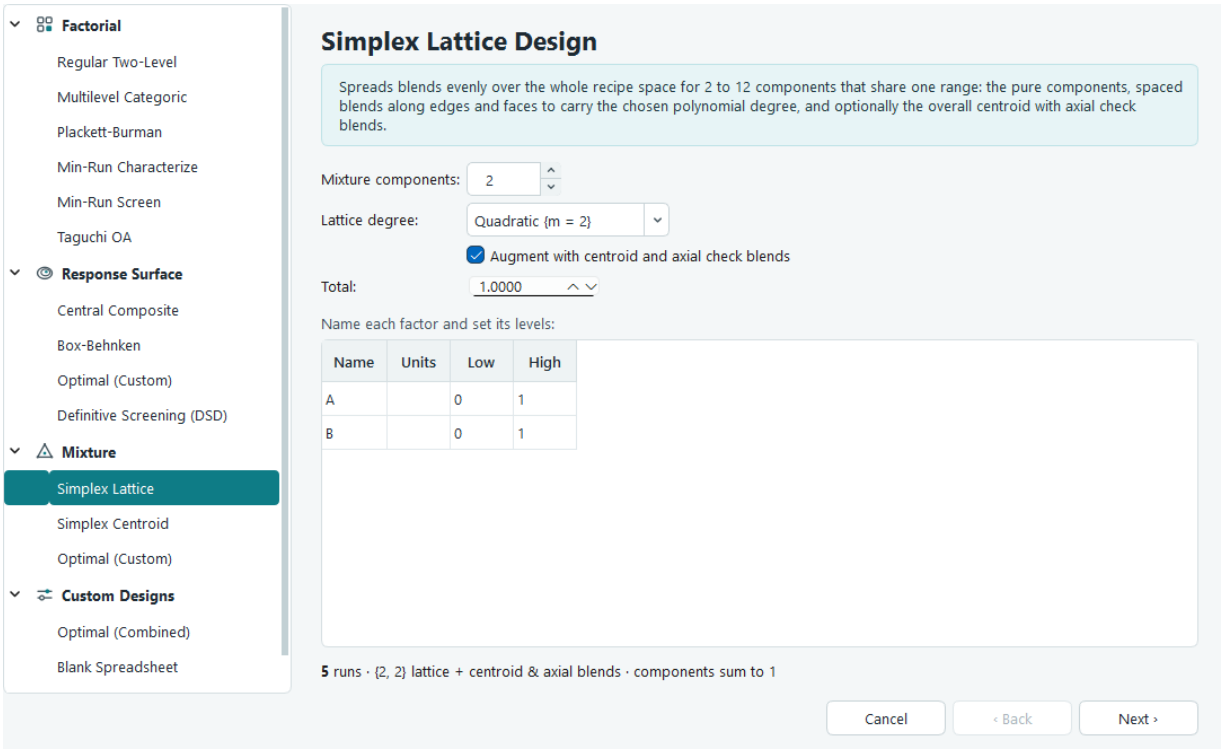
The difference is the constraint. In a factorial you can turn one factor up while everything else stays put. In a blend you cannot: if you raise lactose from 30% to 40%, that extra 10% has to come out of the other two ingredients. The proportions are not free to move on their own, they always sum to the total. Because of that, a mixture design does not put the ingredients on a square grid. It spreads the blends across a triangle whose corners are the pure components, and it fits a model that respects the sum.

> **Total, not level.**
> A mixture factor is a proportion of the blend, so its meaningful range runs from absent (0) to the whole blend (the total). You set the total once, and every run the software builds respects it.

Build the design

A **simplex lattice** is the natural first choice. It places runs at the pure components, at evenly spaced blends along the edges of the triangle, and, when you ask for it, at the centre and a few check blends inside. That set of blends is exactly what the mixture model needs.

1. From the welcome screen choose **New Design** , open the **Mixture** family and select **Simplex Lattice** . The panel on the right explains what the design covers before you commit to it.



The Simplex Lattice page. The run count on the right updates as you change the component count, the degree and the augmentation.

2. Set **Mixture components** to 3. Simplex lattice supports 2 to 12 components.
3. Set the **Lattice degree** to **Quadratic {m = 2}** . The degree controls how many blends sit along each edge, and therefore which model the design can support. Quadratic is the usual working choice: it lets curvature and two component blending show up.
4. Leave **Augment with centroid and axial check blends** ticked. The plain lattice can fit the model but leaves nothing over to test it. The extra centre and axial blends give you runs to check the fit and estimate lack of fit.
5. Set the **Total** to 100, since you are working in percent of the blend. The default total is 1, which suits fractions.
6. Give each component a name and a unit on the factor rows, for example `Lactose` , `MCC` and `Maize Starch` , all in `% w/w` . Add one response, `Disintegration Time` in `sec` . Press **Finish** to build the run sheet.

Internally the component letters are A, B and C, and the labels are the names you typed, so the run sheet headers read `A:Lactose (% w/w)`, `B:MCC (% w/w)` and `C:Maize Starch (% w/w)`.

The blends the software builds

A quadratic three component lattice has six blends: the three pure components and the three fifty to fifty edge blends. Augmentation adds the overall centroid and three axial check blends, for thirteen runs in all... no: ten runs in all. Here they are as percentages of the blend.

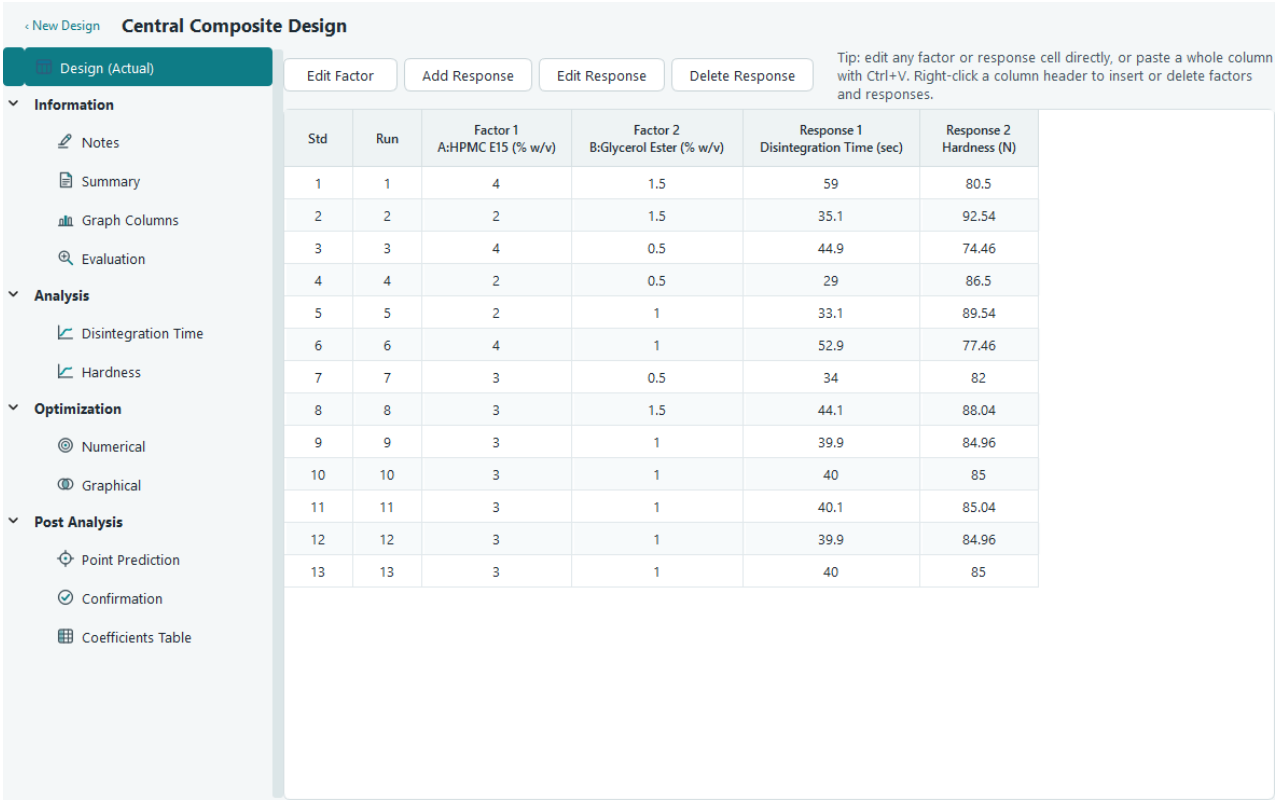
Blend	A: Lactose	B: MCC	C: Maize Starch
Pure A	100	0	0
Pure B	0	100	0

Blend	A: Lactose	B: MCC	C: Maize Starch
Pure C	0	0	100
Edge A and B	50	50	0
Edge A and C	50	0	50
Edge B and C	0	50	50
Centroid	33.3	33.3	33.3
Axial toward A	66.7	16.7	16.7
Axial toward B	16.7	66.7	16.7
Axial toward C	16.7	16.7	66.7

Every row sums to 100, which is the point. The pure components pin down the corners, the edge blends test how each pair works together, and the centroid and axial blends probe the middle where all three ingredients are present. The *Run* column randomises the order you should mix and measure them.

Enter your results

Run the blends, measure the disintegration time for each, and type the numbers into the response column of the run sheet. You can paste a whole column from a spreadsheet with `Ctrl` + `V` and undo a slip with `Ctrl` + `Z`.

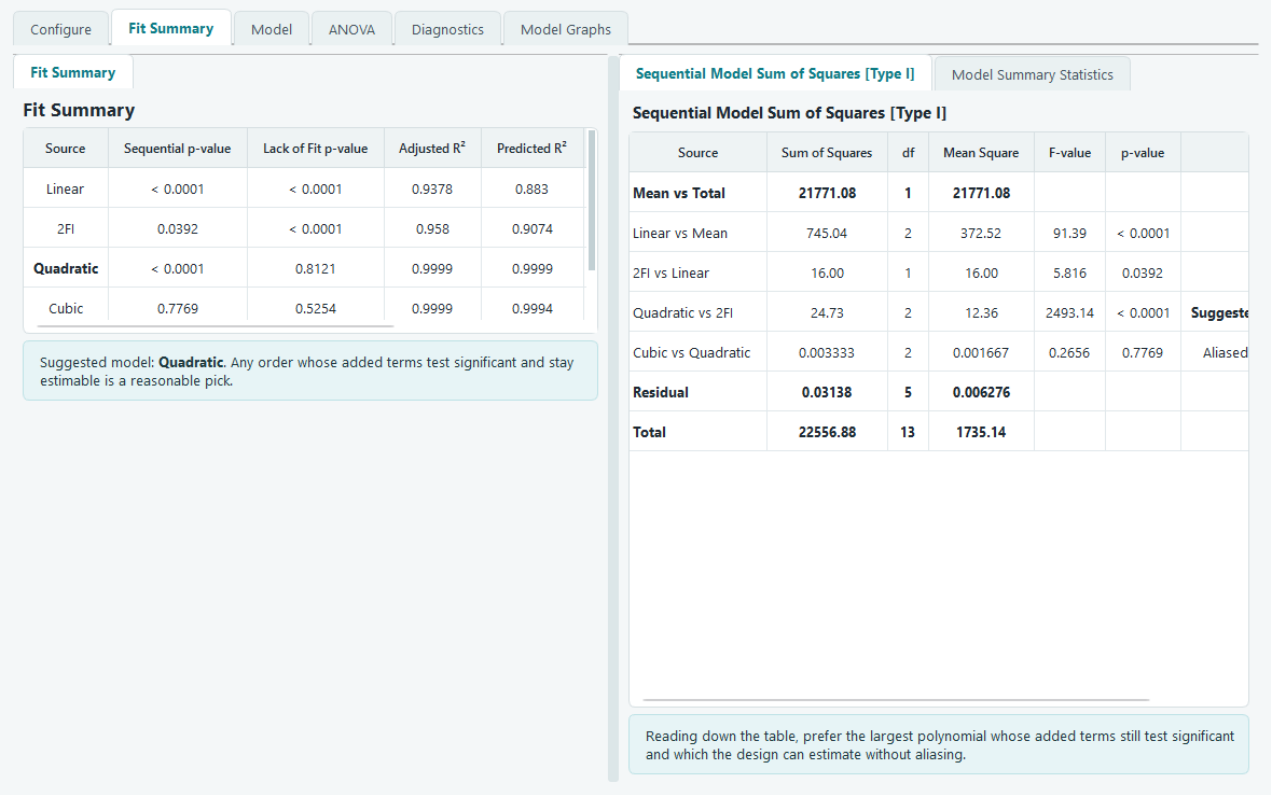


The run sheet layout. This screenshot comes from the two factor central composite example, not from the blend above, so use it to recognise the screen rather than to read the values. Your mixture sheet will have three component columns and one response.

Fit the mixture model

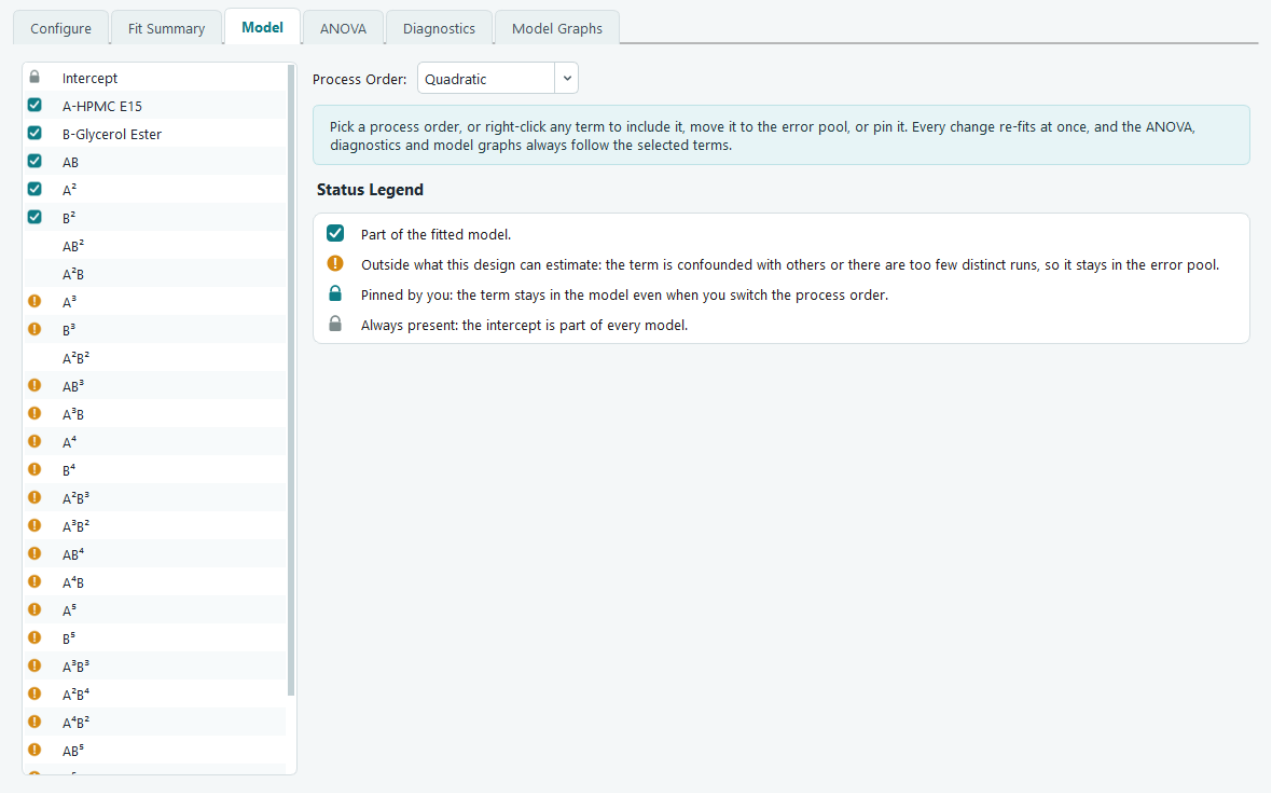
Open the response under **Analysis** and work along the tabs. The **Fit Summary** tab compares model orders and points you at the one the data supports. For a mixture the model carries a linear term for each component

and, in the quadratic model, a cross term for each pair of components that captures how two ingredients blend. Because the components always add up to the same total, one component term cannot be estimated on its own and is reported as *Aliased*. That happens for every mixture design and costs you nothing: the predictions, the graphs and the optimum are unaffected. See how a mixture model is fitted for the reason.



Fit Summary, shown here for the central composite example. It weighs the linear model against the quadratic one and flags the order that fits without over reaching.

On the **Model** tab you can see and adjust the terms. This is also where the mixture constraint becomes visible in the software. Because the three components always add up to the total, one of them carries no information the other two do not already give: once you know lactose and MCC, maize starch is fixed. OptiGein detects that redundancy and marks the extra term as one this design cannot estimate, so it is left out of the fit rather than causing a divide by zero.



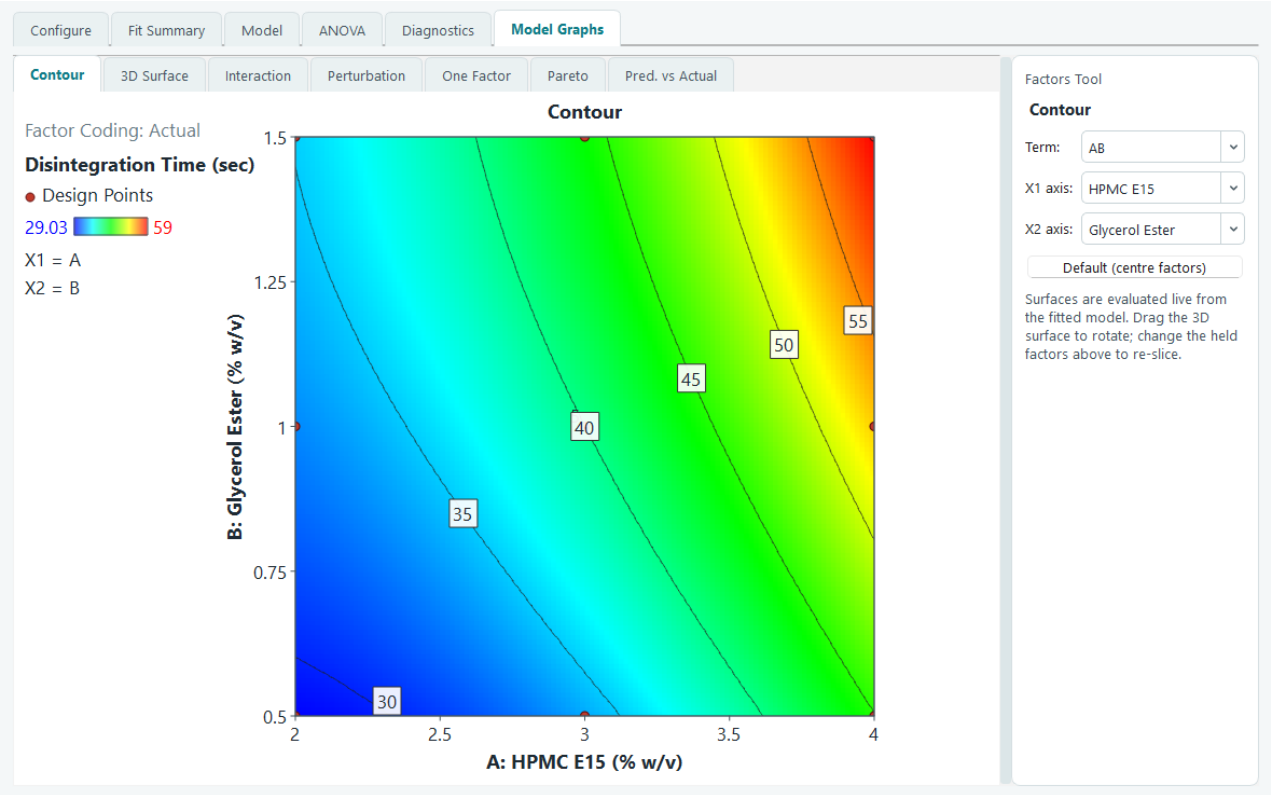
The Model tab. A term the design cannot estimate is flagged with a warning icon and the note that it is beyond what this design can estimate.

The coefficients read differently here.
In classical mixture theory the model has no intercept, and each linear coefficient is the predicted response when that ingredient is the whole blend, that is, the pure component. OptiGein does not print that canonical form. It fits the equivalent slack variable polynomial: it keeps an intercept and drops one component as the redundant term described above. The fitted surface is the same either way, but the coefficients table and the equation you see will show an intercept and one fewer component. Do not read a single coefficient as a pure component response. To get the predicted response at a pure component, use Point Prediction, described below.

The **ANOVA** and equation panels behave as they do for any response. The coded and actual equations list the intercept and the estimable terms, and each table on the page carries a short plain language note under it that says what the numbers mean for your blend.

Read the surface

The **Model Graphs** tab draws the fitted response. It is worth being clear about what you see, because a mixture is often shown as a shaded triangle in textbooks. OptiGein draws the same rectangular contour and 3D surface it uses for a response surface design: two components go on the axes across their ranges, and the remaining component is held at a value you set with the sliders. It does not draw a triangular, three axis mixture plot.

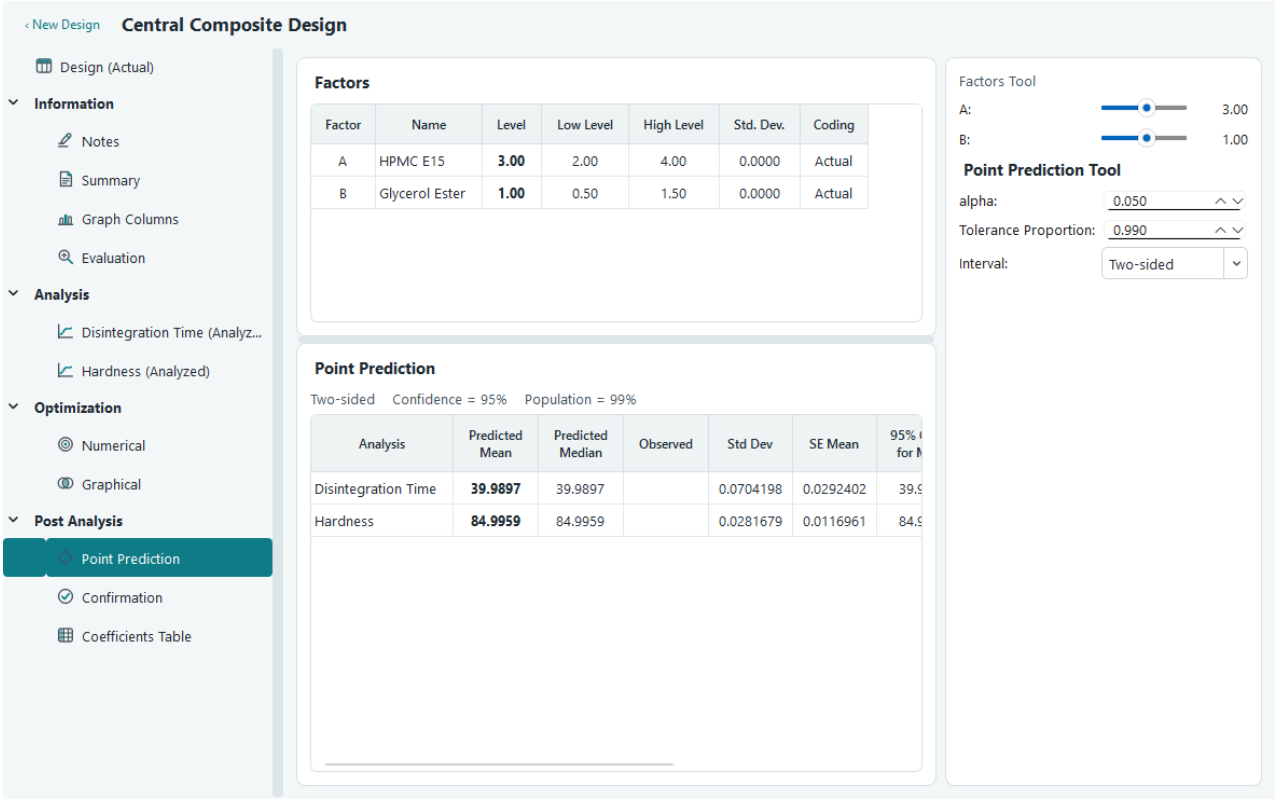


The contour view, shown here for the central composite example. For a blend, two components sit on the axes and the third is held at the slider value.

To read it as a recipe, remember the sum. If lactose is on the horizontal axis, MCC on the vertical axis and maize starch is held at, say, 20%, then only the blends where lactose plus MCC make up the other 80% are real recipes. Those lie along one diagonal line of the plot. Slide the held component to sweep that line across the triangle. When you want the response for one exact blend rather than a slice, use Point Prediction instead, which is not tied to a two component view.

Predict any blend, and the pure components

Under **Post Analysis**, open **Point Prediction**. Type the proportions of a blend you care about and the software returns the predicted disintegration time with a confidence interval. This is how you recover the pure component values that the classical coefficients would have given you: set one component to 100 and the other two to 0, and read the prediction. Do the same at any interior blend to compare recipes before you commit to a confirmation run.



Point Prediction, shown for the central composite example. Enter a blend, including a pure component, and read the predicted response and its interval.

Compare recipes side by side.
The **Coefficients Table** under Post Analysis gathers the fitted terms and their standard errors in one place. It is useful for reporting, but for interpreting a blend the Point Prediction reading is clearer than any single coefficient.

What to read next

- Mixture designs for the simplex centroid and optimal mixture options, and when to prefer each.
- Fit summary, model and ANOVA for how the model tabs work in detail.
- Prediction and confirmation for Point Prediction and planning the runs that confirm your chosen blend.

Part 7

Reference

Chapter 21

Saving, printing and exporting

Your experiment is worth keeping and worth sharing. This page covers the design file you save to disk, and every way OptiGein Design hands your work to other programs: a spreadsheet, a printer, Word and PowerPoint.

Saving your work as a design file

OptiGein Design stores each experiment in a single file with the extension `.ogdx`. One file holds the whole project, so there is nothing else to keep track of. Use **File › Save** (`Ctrl` + `S`) once you have built or opened a design. The first time you save, the software asks where to put the file and adds the `.ogdx` ending for you if you leave it off. After that, Save writes straight back to the same file. **File › Save As** (`Ctrl` + `Shift` + `S`) makes a fresh copy under a new name, which is handy before you try a different model or a what-if change.

A design file is a complete snapshot, not just the run sheet. When you save, it records:

- the design type and its structure, including any blocking and, for fractional designs, the defining relation and alias information;
- every factor with its letter, its name, its units and its low and high settings, and every response with its name and units;
- all runs in both standard and run order, the coded values, and any response values you have measured and typed in;
- your notes for the design;
- the analysis choices you made for each response, such as the model order, any response transform, and which terms you kept in the model.

Because the analysis choices travel with the file, reopening a design puts you back exactly where you left off, not at a blank slate. The file itself is plain text in a readable JSON layout, which keeps it small and easy to archive alongside your lab records.

> **Unsaved changes are flagged.**
> The window title shows an asterisk before the file name while there are edits you have not saved yet, and reads *Untitled* for a design you have never saved. If you close a design or the program while changes are pending, OptiGein Design asks whether to save them first, so you cannot lose work by accident.

Opening a design again

Use **File › Open Design** (`Ctrl` + `O`) to browse for an `.ogdx` file. The files you worked on most recently are quicker to reach: they are listed under **File › Open Recent** and on the welcome screen, so a click reopens them. If a recent file has since been moved or deleted, the software tells you and offers to drop it from the list.

You do not have to start from inside the program at all. If you allowed it during installation, the `.ogdx` type is registered with Windows, so **double clicking a design file in File Explorer opens it directly in OptiGein**

Design. Design files also carry the application icon so they are easy to spot in a folder. The installer offers this as the option *Open .ogdx design files with OptiGein Design by default*, which is switched on unless you clear it.

Exporting the run sheet to a spreadsheet

Sometimes you want the raw table in Excel, or in a colleague's tool that does not read `.ogdx` files. **File › Export Run Sheet to CSV** writes the run sheet as a comma separated file that any spreadsheet opens. This command becomes available as soon as a design is open, and it suggests a file name based on your design file so the two sit together.

The exported table starts with a *Std* column (standard order) and a *Run* column (the randomised run order), and a *Block* column when the design is blocked. Then come the factor columns, headed the same way you see them in the run sheet, for example `A:HPMC E15 (% w/v)`, and finally the response columns, such as `Disintegration Time (sec)` and `Hardness (N)`. Rows are written in run order, the order you would actually perform the experiments. Text that contains a comma or a quotation mark is quoted correctly, and the file is written in UTF-8, so units and special characters survive the trip.

> **CSV is the data, not the analysis.**
> A CSV file holds the factor settings and measured responses only. It is meant for sharing numbers or importing them elsewhere. To keep your models, graphs and optimization work, save the `.ogdx` file as well.

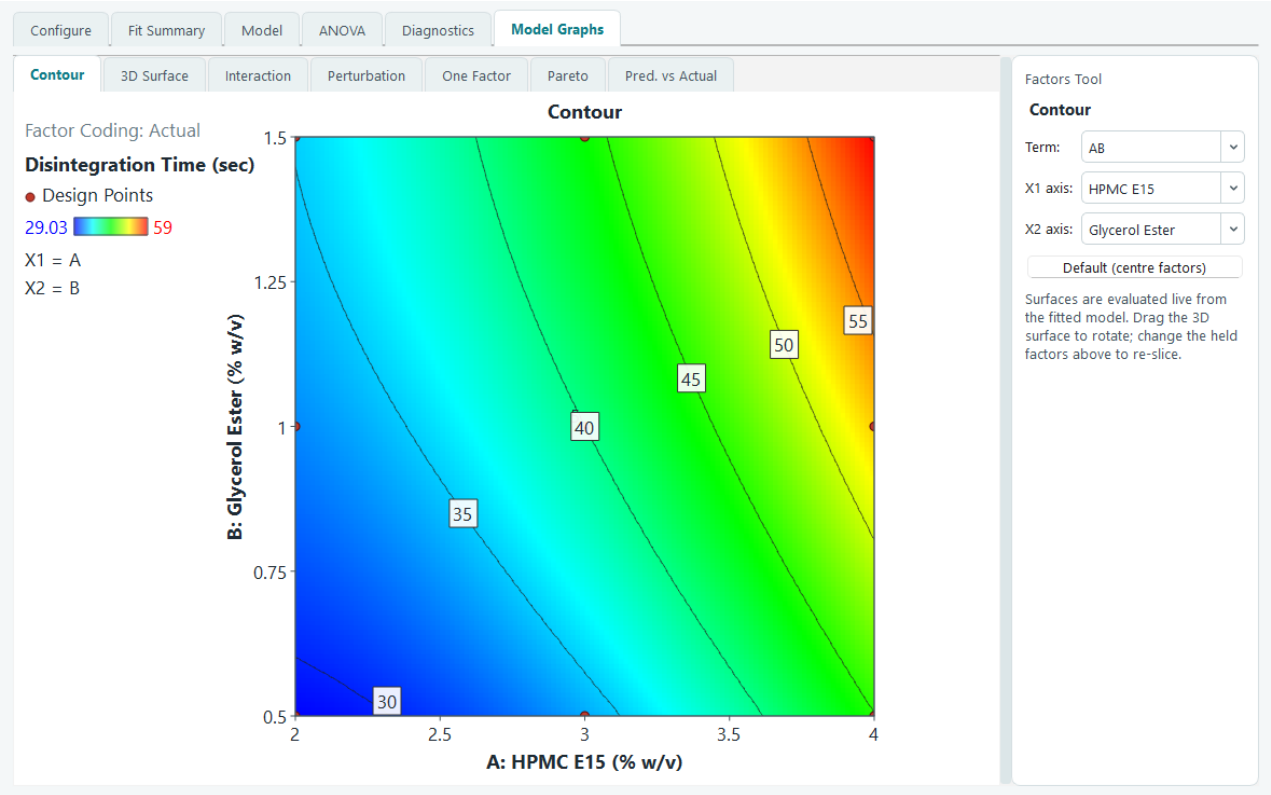
Printing and print preview

Every screen in the workspace can go straight to paper or to a PDF printer. **File › Print** (`Ctrl` + `P`) prints the view you are currently looking at, whether that is the run sheet, an ANOVA table, a diagnostic plot or a contour graph. The page is rendered at high resolution and scaled to fit the paper, so it stays sharp. **File › Print Preview** shows the same output first, so you can check the fit and pick a printer or page size before committing.

Copying and exporting a graph

Any graph in the software carries its own menu. **Right click the graph** and you can send it out in a few ways:

Menu item	What it does
Copy	Places the graph on the clipboard as a high resolution image. Paste it into almost any program with `Ctrl` + `V`.
Export › Export Graph to Word	Inserts the graph into your open Word document, or a new one if none is open.
Export › Export Graph to PowerPoint	Adds the graph to the current PowerPoint slide, or a new presentation, sized to sit neatly on the slide.
Export › Export Graph to File	Saves the graph as a PNG or JPEG image at a location you choose.



A contour plot. Whichever way you export it, the graph is redrawn cleanly first, so it looks the same in your report as it does on screen.

There is an important detail behind all of these. A graph is not simply screenshotted from the window. Before it is copied or exported, OptiGein Design **re-renders it at a fixed, readable size and a high resolution, independent of your window size**. So a graph copied from a small or cramped window still comes out with full axis titles, no overlapping labels and crisp lines in the finished document. The image also carries its true resolution, so Word and PowerPoint place it at a sensible size rather than blowing it up.

Copying and exporting a results table

The tables on the analysis and optimization screens work the same way. **Right click a results table** to reach these commands:

Menu item	What it does
Copy	Copies the cells you have selected. Paste them into a spreadsheet or document and they land as a real table, not run-together text.
Copy With Headings	The same, but includes the column headings above the copied cells.
Export Table to Word	Sends the whole table, with headings, into Word as a formatted table you can edit.
Export Table to PowerPoint	Places the whole table, with headings, onto a PowerPoint slide as a formatted table.

Copy and Copy With Headings put the table on the clipboard in a rich form as well as plain text, so pasting into Word, Excel or PowerPoint gives you a genuine table with rows and columns, ready to restyle. The two Export commands go a step further and build the table directly in a formatted layout, with a shaded heading row, so numbers from your ANOVA or coefficients table drop into a report already looking tidy.

Word and PowerPoint export needs Microsoft Office.
The Export to Word and Export to PowerPoint commands drive Office directly, so they work on Windows with Word or PowerPoint installed. If Office is not present, or you are on another operating system, use *Copy* and paste into your document instead. For graphs you also have *Export Graph to File* to save a PNG or JPEG. The software tells you when direct Office export is not available and points you to these alternatives.

What to read next

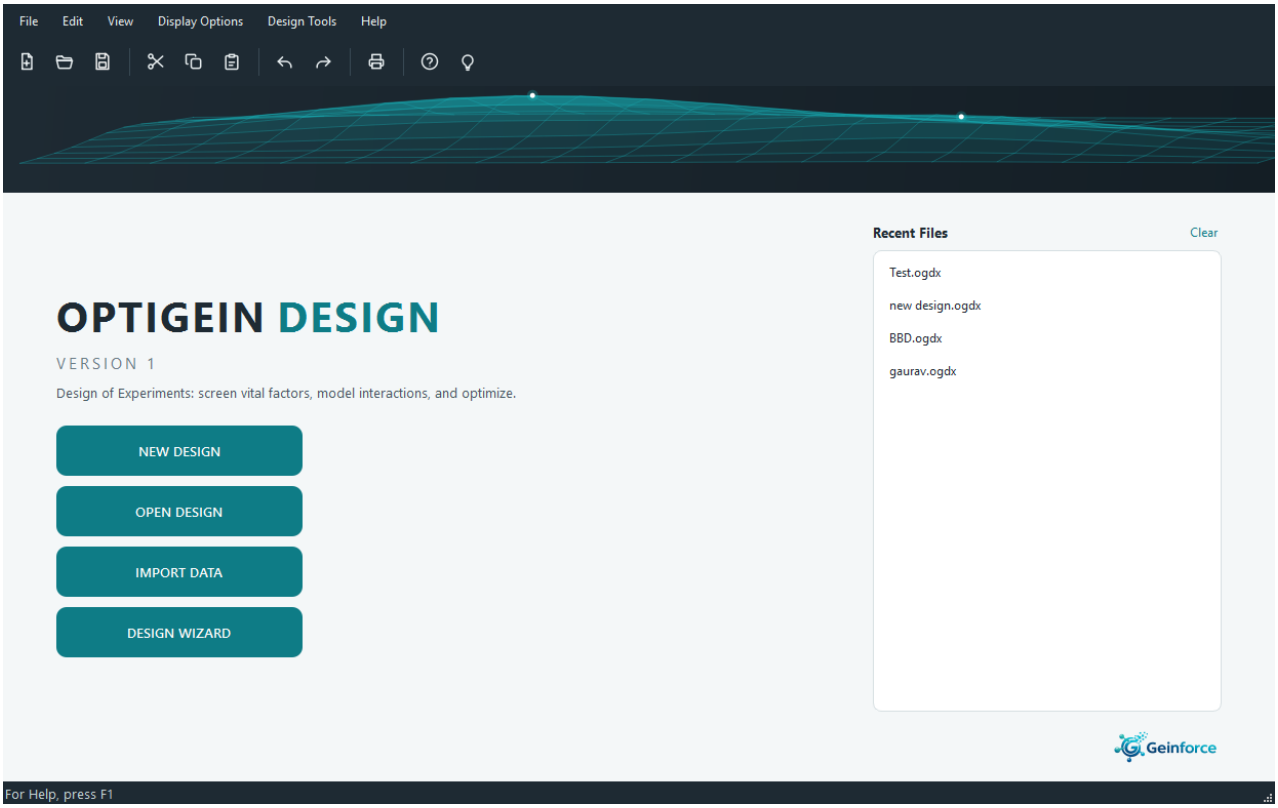
These commands live in the File menu and the right click menus. For the full list of menu items and their shortcuts, see Menus and shortcuts. For what each analysis table and graph means before you export it, see Fit summary, model and ANOVA and Model graphs.

Chapter 22

Menus and shortcuts

A complete reference to the menu bar, the toolbar and the keyboard shortcuts. Keep it open beside the application if you like knowing exactly what a command does before you use it.

The menu bar and toolbar sit above every screen and stay in the same place whether you are on the welcome screen, building a design or reading an ANOVA table. Some commands only make sense once a design is open. Those stay greyed out until you build or open one, so you can never trigger them by accident.



The menu bar and toolbar run along the top of the window and are available on every screen.

> **Reading the shortcut columns.**
> A blank shortcut cell means the command has no default key. On Windows the Exit command closes the window the same way the title-bar close button does, so it is left without a key of its own.

File menu

Everything to do with whole design files lives here: starting one, opening one, saving your work, and getting the run sheet out as a spreadsheet or onto paper.

Item	Shortcut	What it does
New Design	`Ctrl+N`	Open the design catalogue to start a new design of experiments.
Open Design	`Ctrl+O`	Open a saved `.ogdx` design file from disk.

Item	Shortcut	What it does
Open Recent		A submenu of the files you opened lately. It ends with Clear Recent Files, and reads "(No recent files)" until you have opened something.
Close Design	`Ctrl+W`	Close the current design and return to the welcome screen. You are asked to save first if there are unsaved changes.
Save	`Ctrl+S`	Save the current design. If it has never been saved, this behaves like Save As.
Save As	`Ctrl+Shift+S`	Save the current design to a new `.ogdx` file, adding the extension for you if you leave it off.
Import Data		Open the builder at the Import Data Set screen, where you paste tabular results and assign each column as a factor or a response.
Export Run Sheet to CSV		Write the run sheet to a CSV file, ordered by run number, with the Std and Run columns, a Block column when the design is blocked, then every factor and response.
Print Preview		Show how the screen you are on now will look when printed, scaled to fit the page.
Print	`Ctrl+P`	Print the current screen as a picture, centred and scaled to the page.
Exit		Close OptiGein Design, prompting you to save any unsaved design first.

What gets printed.
Print and Print Preview capture whatever screen is showing, so switch to the run sheet, an ANOVA table or a model graph first, then print that view.

Edit menu

The edit commands act on whatever has focus at that moment: a cell in the run sheet, a text box in a dialog, or a table of results. That is why the same Copy command works on a run-sheet selection and on an ANOVA table. In read-only result tables, Cut and Copy both place the selected cells on the clipboard as tab-separated text you can paste into a spreadsheet.

Item	Shortcut	What it does
Undo	`Ctrl+Z`	Undo the last edit in the active editor or run sheet.
Redo	`Ctrl+Y`	Redo the last undone edit.
Cut	`Ctrl+X`	Cut the selection to the clipboard. In read-only tables it copies instead.

Item	Shortcut	What it does
Copy	`Ctrl+C`	Copy the selection to the clipboard.
Paste	`Ctrl+V`	Paste from the clipboard into the focused editor or run-sheet cells.
Delete	`Del`	Clear the selected cells, or delete the next character in a text box.
Select All	`Ctrl+A`	Select everything in the active editor or table.

View menu

The View menu controls the window itself, not your data. The Toolbar and Status Bar entries are checkboxes that show or hide those bars, and Full Screen is a toggle you can turn on and off with the same key.

Item	Shortcut	What it does
Toolbar		Show or hide the toolbar. The tick follows the toolbar's current state.
Status Bar		Show or hide the status bar along the bottom of the window.
Full Screen	`F11`	Fill the whole screen. Press again to return to a normal window.
Reset Window Layout		Restore a sensible default window size, leave full-screen mode, and bring back the toolbar and status bar.

Display Options menu

Item	Shortcut	What it does
Show Tip on Startup		A checkbox that records whether you want the Design of Experiments tip shown at startup. Your choice is remembered between sessions, and you can read a tip anytime from Help, Tip of the Day.

Design Tools menu

This menu gathers the three ways to begin. New Design and Import Data are the same commands as in the File menu, repeated here for convenience. The Design Wizard is unique to this menu.

Item	Shortcut	What it does
New Design	`Ctrl+N`	Open the design catalogue, the same as File, New Design.
Design Wizard		Answer a few questions about your goal, your factors and your run budget, and let OptiGein recommend and set up a design for you.

Item	Shortcut	What it does
Import Data		Open the builder at the Import Data Set screen, the same as File, Import Data.

Help menu

Help gathers the built-in guidance, the ways to reach Geinforce Technology, the two commands that manage your license on this computer, and the check for a newer version.

Item	Shortcut	What it does
Help Contents	F1	Show a short guide to working through an experiment, from building a design to confirming the result.
Keyboard Shortcuts		Show the list of keyboard shortcuts in a small window.
Tip of the Day		Show a Design of Experiments tip, with a Next Tip button to read more.
Visit Geinforce Website		Open geinforce.com in your browser.
Contact Support		Show the support email and phone number for Geinforce Technology.
License Information		Show the license stored on this computer: its status, type, expiry date, who it is licensed to, and how many device slots are in use.
Deactivate This Device		Release this computer's activation slot so the license can be used elsewhere. OptiGein Design closes afterwards and will ask for activation next time it starts.
Check for Updates		Ask the Geinforce server whether a newer version exists. If one does, the window shows what changed and offers to download and install it. See Keeping the software up to date.
About OptiGein Design		Show the version number and company details.

Deactivate frees a slot.
Use Deactivate This Device before moving your license to another machine. It tells the license server to free the slot and removes the license from this computer, so keep your license key somewhere safe if you plan to reactivate later.

Toolbar buttons

The toolbar is a shortcut to the commands people reach for most often. Every button repeats a menu item, so hovering over one shows its name and, where it has a key, its shortcut. Save is greyed out until a design is open.

Button	Command	Shortcut
New	File, New Design	`Ctrl+N`
Open	File, Open Design	`Ctrl+O`
Save	File, Save	`Ctrl+S`
Cut	Edit, Cut	`Ctrl+X`
Copy	Edit, Copy	`Ctrl+C`
Paste	Edit, Paste	`Ctrl+V`
Undo	Edit, Undo	`Ctrl+Z`
Redo	Edit, Redo	`Ctrl+Y`
Print	File, Print	`Ctrl+P`
Help	Help, Help Contents	`F1`
Tip of the Day	Help, Tip of the Day	

You can hide the whole bar from View, Toolbar if you want more room for a graph or a wide run sheet, and bring it back the same way.

Chapter 23

Glossary

Plain definitions of the design and statistics terms you meet across OptiGein Design. Each entry points to the page that explains it in context, so you can read a full definition here and the worked detail there.

How to read the letters below.
Terms are grouped by first letter. Standard field vocabulary (ANOVA, p-value, desirability) is used as is; where the software gives a term a particular meaning on screen, the definition matches what you actually see.

A

Term	What it means
Adequate precision	A signal to noise measure. It compares the range of predictions across the design points against the average prediction error. A value above about 4 means the model is good enough to navigate the design space. See Fit summary, model and ANOVA.
Adjusted R squared	R squared corrected for the number of terms in the model, so adding a term that does not really help no longer inflates the score. See Fit summary, model and ANOVA.
Alias	When a design cannot tell two effects apart, they are aliased: their estimates add together and the software reports them as one. The Alias Matrix lists them. See Evaluating a design.
ANOVA	Analysis of variance. The table that splits the variation in a response into the part each model term explains and the part left over, then tests whether each part is large enough to matter. See Fit summary, model and ANOVA.
Axial point	In a central composite design, a run placed out along one factor axis, on the face of the factorial box or beyond it, used to estimate curvature. Also called a star point. See Response surface designs.

B

Term	What it means
Box-Behnken	A response surface design that places runs at the midpoints of the edges of the factor space plus centre points, so it fits a quadratic model without any extreme corner settings. See Response surface designs.
Box-Cox	A plot that suggests whether a power transform of the response would steady the spread of the residuals, and which power to use. See Diagnostics.

C

Term	What it means
Centre point	A run with every numeric factor set to the middle of its range. Repeated centre points measure pure error and reveal curvature. The example design carries five of them. See Response surface designs.
Coded factor	A factor rescaled so its low level is minus one and its high level is plus one, with the centre at zero. Coefficients on coded factors can be compared directly because they share a scale. See Fit summary, model and ANOVA.
Confidence interval	A range that is likely to contain the true mean response at a given set of factor settings. The Point Prediction page reports it. See Prediction and confirmation.
Cook's distance	An influence measure for a single run. A large value means dropping that run would noticeably change the fitted model. See Diagnostics.
Corrected total	The total variation in the response about its mean, before the model explains any of it. It is the bottom line of the ANOVA table and equals the model plus residual sums of squares. See Fit summary, model and ANOVA.
C.V. percent	The coefficient of variation: the residual standard deviation written as a percentage of the mean response, a quick read on the relative noise in the experiment. See Fit summary, model and ANOVA.

D

Term	What it means
Degrees of freedom	The number of independent pieces of information behind a quantity, roughly the number of runs minus the number of things estimated from them. See Fit summary, model and ANOVA.
Desirability	A score from 0 to 1 that says how well one set of factor settings meets a goal. The optimizer combines the desirability of every response into one overall score and searches for its highest value. See Numerical optimization.
DFBETAS	For one run and one model term, how far that run shifts the term's coefficient, measured in standard errors. It shows which run drives which coefficient. See Diagnostics.
DFFITS	How far a single run shifts its own fitted value, in standard errors. Another way to spot a run with heavy influence. See Diagnostics.

F

Term	What it means
Factor	An input you set on purpose and vary across the experiment, such as a concentration or a temperature. Factor A carries the letter; its label is the name you type, so a column reads like A:HPMC E15 (% w/v). See The design workspace.

Term	What it means
Fold-over	A second block of runs built by reversing the signs of a fractional factorial, added to break aliases and raise the resolution of the combined design. See Factorial designs.
Fraction of design space	The FDS graph reads the fraction of the design region along the horizontal axis against the largest prediction error over that fraction, so you can see how evenly the design predicts. See Evaluating a design.

H

Term	What it means
Hierarchy	The rule that if a model keeps a higher order term it should also keep the lower order terms inside it, for example keeping A and B whenever AB is present. The software preserves hierarchy when it builds an equation. See Fit summary, model and ANOVA.

I

Term	What it means
Interaction	When the effect of one factor depends on the level of another. An interaction term such as AB captures that dependence. See Fit summary, model and ANOVA.

L

Term	What it means
Lack of fit	A test that compares the model's error against the pure error from replicated runs. A significant lack of fit means the chosen model shape misses something real in the data. See Fit summary, model and ANOVA.
Lambda	The power in a Box-Cox transform. Lambda of 1 is no transform, 0 is a log and 0.5 is a square root. See Diagnostics.
Leverage	How much a single run's own settings pull the fitted model toward its response. Leverage runs from 0 to 1, and a value near 1 means the run largely fits itself. See Diagnostics.

M

Term	What it means
Main effect	The average change in a response as one factor moves from its low to its high level, on its own. See Factorial designs.
Mixture	An experiment where the factors are the proportions of the parts of a blend, so they must add up to a fixed total, usually 100 percent. See Mixture designs.

Term	What it means
Model order	How complex the fitted equation is: linear (main effects only), two factor interaction, quadratic, or cubic. See Fit summary, model and ANOVA.

O

Term	What it means
Orthogonal	A design property where the factor columns are uncorrelated, so each effect is estimated independently of the others without inflating its error. See Evaluating a design.

P

Term	What it means
Predicted R squared	A measure of how well the model predicts a run that was left out, built from PRESS. It should stay close to the adjusted R squared; a wide gap warns of overfitting. See Fit summary, model and ANOVA.
Prediction interval	A range likely to contain a single new measurement, not the mean, at a given set of settings. It is wider than the confidence interval. See Prediction and confirmation.
PRESS	Predicted residual error sum of squares. Each run is dropped in turn, the model is refit, and the miss on the dropped run is squared and summed. Smaller is better. See Fit summary, model and ANOVA.
Pure error	The variation between replicated runs at identical settings. It measures the noise of the experiment on its own, with no help from any model. See Fit summary, model and ANOVA.
p-value	The chance of seeing an effect this large if the term truly did nothing. A small p-value, commonly below 0.05, is read as evidence the term matters. See Fit summary, model and ANOVA.

R

Term	What it means
Randomisation	Performing the runs in a shuffled order so that anything drifting over time is not mistaken for a factor effect. The Run column holds this order. See The design workspace.
Replicate	A run repeated at exactly the same factor settings. Replicates supply the pure error estimate. See Which design do I need?.
Residual	The gap between a measured response and the value the model predicts for that run. The diagnostics work almost entirely with residuals. See Diagnostics.
Resolution	A rating of a fractional factorial that says how badly effects are aliased. Resolution V keeps main effects and two factor interactions clear of each other; lower numbers mix them. See Factorial designs.

Term	What it means
Response	An output you measure on each run, such as disintegration time or hardness. You analyse one response at a time. See The design workspace.
Response surface	A smooth fitted model, usually quadratic, that maps the responses across the factor space so you can find peaks, valleys and good operating regions. See Response surface designs.
Run	One experiment: a single row of the run sheet, with its factor settings and the responses you record. The example design has thirteen. See The design workspace.

S

Term	What it means
Scheffe model	The classical polynomial form for mixture data, which drops the intercept and keeps one linear term per component. OptiGein Design fits the equivalent slack variable form instead: it keeps the intercept and aliases away one component term. Both give identical predictions. See Mixture designs.
Screening	An early stage that uses a small design to sift many factors and find the few that actually matter, before studying them in detail. See Factorial designs.
Sequential sum of squares	The Type I breakdown that adds model terms in order and credits each with the extra variation it explains on top of the terms already in. It drives the model order suggestion on Fit Summary. See Fit summary, model and ANOVA.
Simplex	The triangular, or higher dimensional, region of allowed blends in a mixture problem, where the components add up to the total. Simplex-lattice and simplex-centroid designs place runs across it. See Mixture designs.
Standard order	The tidy, unrandomised order in which the design is written out on paper, before the run order shuffles it. See The design workspace.
Studentized residual	A residual divided by its own standard error so all residuals share one scale. The internally studentized version uses every run; the externally studentized version leaves the run out first, which makes outliers stand out. See Diagnostics.

T

Term	What it means
Tolerance interval	A range expected to contain a stated fraction of the whole population of future results, at a stated confidence. The Confirmation page reports it. See Prediction and confirmation.
Transform	A mathematical reshaping of the response, such as a log, square root or inverse, applied before fitting to steady the variance or straighten a relationship. See Diagnostics.

Term	What it means
Two factor interaction	An interaction between exactly two factors, written like AB. It is the simplest kind of interaction and the first curvature a factorial can pick up. See Fit summary, model and ANOVA.

V

Term	What it means
VIF	Variance inflation factor. It says how much a term's variance is enlarged because that term overlaps with the others. A VIF of 1 means the term is estimated independently. See Evaluating a design.

Cannot find a term.
If a word on screen is not listed here, it usually appears with its own short explanation right under the table or graph that uses it. The analysis and diagnostics pages carry the most of these notes.

Chapter 24

Troubleshooting

Most of what looks like a fault is the software telling you something true about your license or your data. This page lists the messages you are most likely to see, what causes each one, and the quickest way past it.

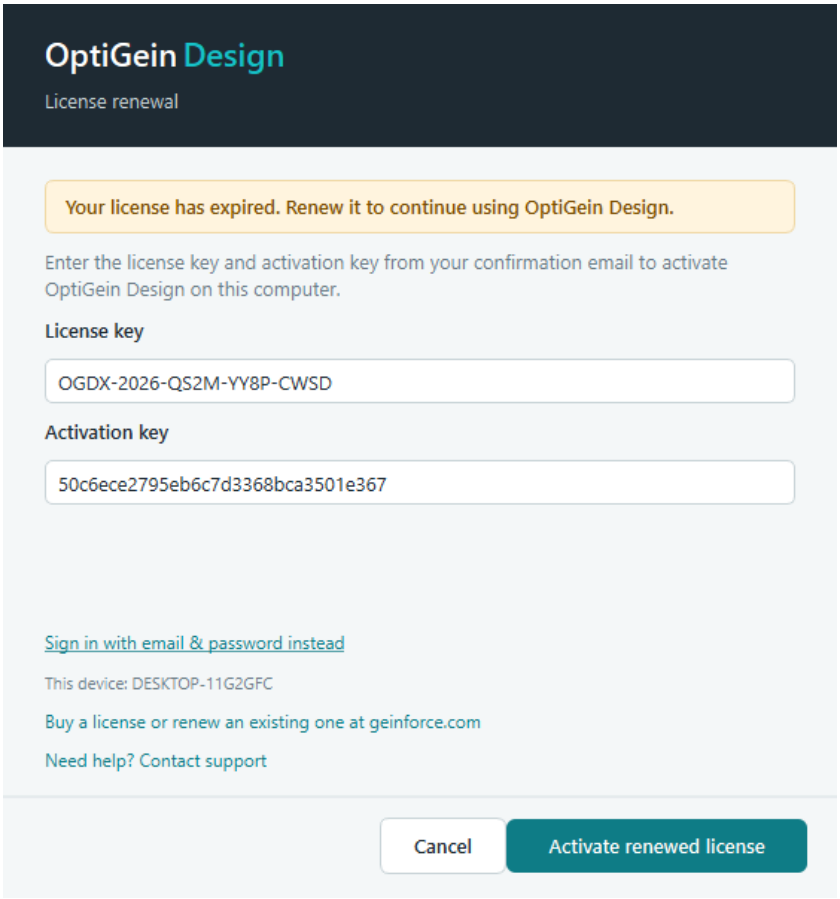
The app will not start

Every time it opens, OptiGein Design reads the license stored on the computer. It needs no internet connection to do that. When there is no stored license, or its period has ended, the activation screen appears instead of the workspace. The app never runs unlicensed, so the way forward is always to activate or renew, not to bypass the screen.

The activation screen says your license has expired

Your license has a fixed end date. Once that date passes, the app stops at the renewal screen, whether or not you are online. A license can also lapse while the app is open: OptiGein Design re-checks the stored license every ten minutes, and if the date has passed it asks you to renew there and then.

Renew the license at geinforce.com, then open the activation screen it presents. Because this is a renewal, the screen is already filled in with your keys, so you press **Activate renewed license** to continue. Your saved designs are untouched; only the license needs refreshing.



The renewal screen, prefilled with your stored keys. Renew at geinforce.com first, then activate.

It asks for activation again on a computer that was already activated

There are two usual causes. The first is an old version. Before 1.4.5 the software included the network adapter in the way it recognised a computer, so joining another Wi-Fi network, using a mobile hotspot, plugging in a network cable or connecting a VPN could make an activated computer look like a new one. Version 1.4.5 recognises the computer by its operating system installation instead, and it keeps an activation that already works. If you are on an older version, install the current one as described in Keeping the software up to date.

The second cause is that the stored license is gone: you chose Deactivate This Device, the operating system was reinstalled, or the user profile was reset. Activate again with the same keys or the same account. If the server answers that the maximum number of devices is reached, the old entry for this computer is still holding a slot; contact support and we will release it.

"Maximum activations reached"

Every license covers a set number of devices. When you try to activate one more than the license allows, the server refuses and the screen reads that the license is already active on the maximum number of devices. The activation summary of a working install shows this as, for example, device 3 of 3.

Free a slot on a machine you no longer use: open that machine, go to **Help › Deactivate This Device**, and confirm. That releases the slot on the server and clears the license from that computer. You can then activate the new machine. If the old machine is gone or will not start, contact support and we will release the slot for you.

The app cannot reach the activation server

This only matters at the moments that talk to the server: activating, activating a renewed license, deactivating a device and checking for updates. An activated copy does not contact the server to start, so being offline never stops you working during the license period.

If activation itself cannot get through, the screen reports a connection problem and retries a few times on its own before returning to the form with a message. Check that the computer is online, and that a firewall or proxy is not blocking `geinforce.com`, then press Activate again.

> **Check the clock.**
> The end date of the license is compared with the date and time on your computer. If the app says a license has expired when you know it has not, or that the clock has been set back, correct the date and time and start the app again.

Updates and appearance

Check for Updates says "Update unavailable"

Help › Check for Updates needs the internet. When the update server cannot be reached, or answers with an error, the window reads Update unavailable and gives the reason underneath. Check the connection and try again later. Your installed version keeps working in the meantime; an update is never forced on you part way through your work.

The update downloaded but did not install

What you see	What it means	What to do
The downloaded update failed its integrity check and was discarded	The file that arrived was not identical to the one Geinforce published, usually because the download was cut short. The software refuses to run it.	Choose Check for Updates again to download a fresh copy.
Could not start the installer	The file was downloaded and verified, but Windows would not run it. Security software sometimes blocks this.	The message shows where the file is. Close OptiGein Design and run that file yourself.
The app closed and did not come back	Installing needs administrator approval. If the Windows prompt was declined or timed out, nothing was installed.	Start OptiGein Design from the Start Menu. You are still on the old version, so run Check for Updates again and approve the prompt.

On Linux the update is not installed for you. *Download Update* saves the package in your Downloads folder and shows the `sudo apt install` command to run in a terminal, as described in Keeping the software up to date. If the window shows *Update Now* and then *Could not start the installer*, you have an early 1.5.0 build: download the new package from *My License* on geinforce.com and install it with `sudo apt install ./optigein-design_<version>_amd64.deb`.

Text is hard to read when the computer is in dark mode

From version 1.4.4, OptiGein Design follows the dark or light setting of your operating system, including its dialogs, message boxes and tooltips. Older versions kept light colours in some places and dark ones in others, which left dark text on a dark background. Update to the current version to fix it. If you prefer the light look, switch the operating system to its light mode and the app follows at once, with no restart.

Analysis

Start Analysis is greyed out, or asks to be pressed again

Both of these are the button doing its job, not a fault. After a fit completes, the button disables itself and reads that the analysis is complete and you should change a setting to re-run. Change the polynomial order or the transform on the Configure tab, or edit the term set on the Model tab, and the button comes back. Editing terms on the Model tab re-fits at once, so you often do not need the button at all.

If you go back and change response values in the run sheet after analysing, the saved analysis no longer matches the data. OptiGein Design clears the stale tables and each tab shows that the design data changed and you should open Configure and press Start Analysis to refresh. Do exactly that. If Start Analysis produces an error about too few points instead, check the Configure tab: it reports how many complete data points the response has, and a model needs enough of them to fit.

The suggested model is Mean

Fit Summary scores the linear, two factor interaction, quadratic and cubic models and suggests the one the data supports. When none of them earns a positive score, meaning no polynomial order explains the response better than its own average, the suggestion falls back to **Mean**. In plain terms, nothing you varied moved the response above the noise.

First rule out a data problem: confirm the response column is entered correctly and open Diagnostics to check for a stray point. If the data is sound, the effect may genuinely be small relative to the scatter, or your factor ranges may be too narrow to reveal it. Try a response transformation, widen the ranges in a follow-up design, or add runs to sharpen the estimates. A Mean suggestion is a real and useful result: it tells you this experiment found no signal.

Lack of fit is significant

The lack-of-fit test compares what your model leaves unexplained against the pure error measured from repeated runs. In the worked face-centred design, the five centre points supply that pure error. When the ANOVA note reports lack of fit as significant, some real pattern in the data sits outside the model you fitted, for example curvature that a linear model cannot bend to.

Raise the polynomial order, from linear toward quadratic, or add the specific higher terms the shape needs on the Model tab. A response transformation sometimes removes the misfit in one step. Also check Diagnostics for a single influential point, since one bad run can trip the test. Remember that lack of fit can only be measured when the design has replicate runs to estimate pure error.

A term will not go into the model

On the Model tab a term can carry a warning mark and refuse to join the fit, staying in the error pool. The status legend spells out why: the term is outside what this design can estimate, because it is confounded with other terms or there are too few distinct runs to separate it. In Fit Summary, a whole order that depends on such terms is labelled **Aliased** and can never be the suggested model.

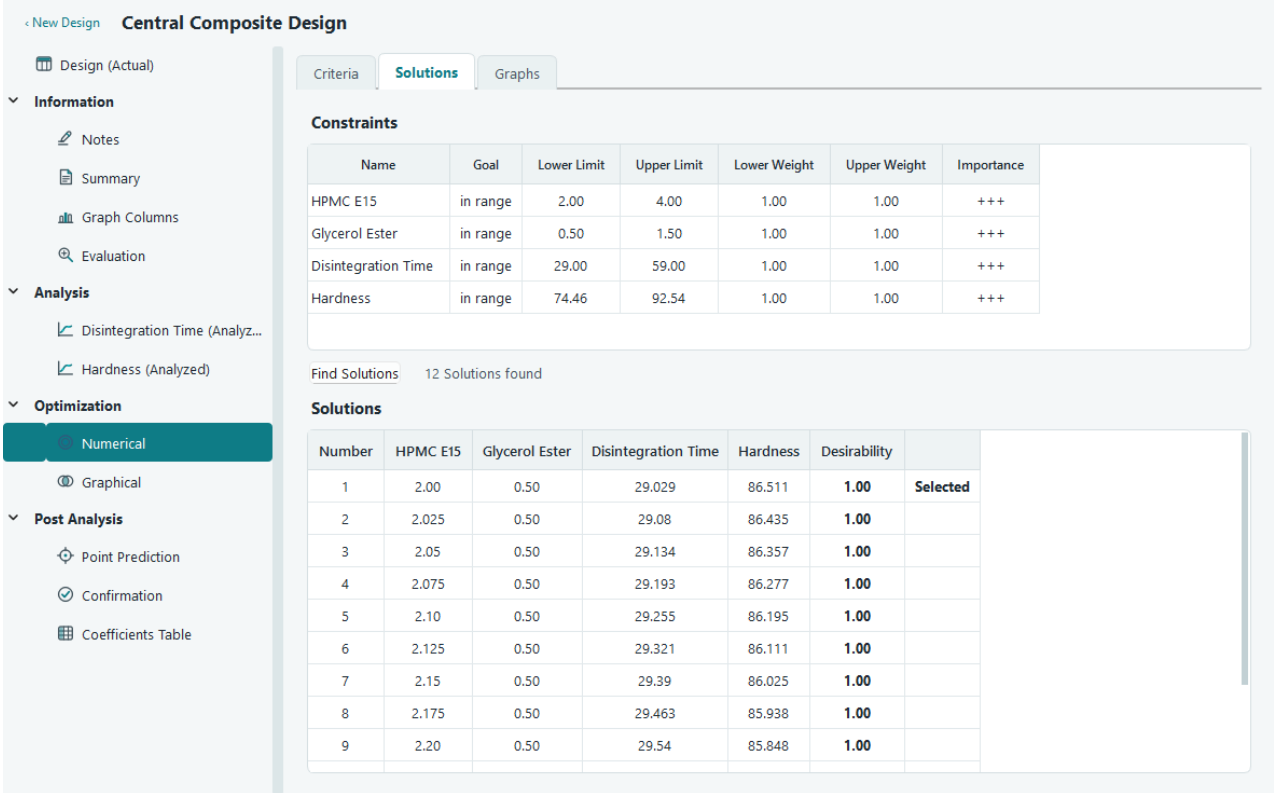
You cannot force an aliased term into a model this design cannot support; the numbers to separate it simply are not there. Either leave the term out and work with what the design can estimate, or build a larger design that resolves it. A screening factorial, for instance, aliases two factor interactions on purpose; if you need those interactions or curvature, move up to a response surface design with enough distinct runs.

Optimization

The optimizer finds no solution, or every desirability is zero

Numerical optimization turns each response goal into a desirability between 0 and 1 and combines them. If any single response with a goal lands outside its limits, its desirability is zero, and one zero drags the overall score to zero as well. When that holds everywhere it searched, the result reads that there is no feasible solution for the given goals. The optimizer still shows its best attempt, so an all-zero desirability means no settings can satisfy every goal at once.

Loosen the limits that are hardest to meet, lower the importance on a less critical goal, or drop a goal to In range so it stops forcing the score to zero. Check that each goal points the way you intend and that any target values sit inside the response range. If you see a message that a response must be analysed first, open its Analysis page and run Start Analysis: the optimizer can only use responses that have a fitted model.



A healthy result lists solutions with non-zero desirability, best first. An all-zero column means the goals conflict.

Graphical optimization shows no yellow region

The overlay plot shades yellow only where every response stays inside its limits at the same time, as the note under the plot states. No yellow means that, for the two factors on the axes and the current settings of the factors you are holding fixed, there is no spot that keeps all responses in bounds together. One-sided limits are stricter still, because the plot judges them against a 95% confidence bound, so it keeps only points that are confidently within spec.

Widen a limit on the Criteria tab, move the sliders for the held factors to a more promising region, or plot a different pair of factors to see whether the sweet spot lies elsewhere. If nothing opens up, the specifications may be incompatible for this system, which is itself worth knowing before you run confirmation experiments.

Graphs and export

Export to Word or PowerPoint does nothing, or reports Office is missing

Right-clicking a graph or a result table offers Export to Word and Export to PowerPoint. These drive a running or newly started copy of the Office application to place the image or table for you. If Office is not installed or cannot start, you get a clear message that Microsoft Word, or PowerPoint, is not installed or could not be started. Direct Office export is available on Windows only; on other systems the app tells you to use Copy or Export to File instead.

You do not need Office to get your graphs out. Use **Copy** to put the image on the clipboard and paste it into any document, use **Copy With Headings** for tables, or use **Export Graph to File** to save a PNG or JPEG. If you do have Office and the export still fails, close any dialog that Word or PowerPoint is waiting on and try again, since the app cannot place content while the other program is blocked.

A graph looks cramped in a small window

Graphs keep a readable minimum size. When the window is too small to hold one at that size, the page scrolls rather than squashing the plot, so axis titles and labels never collapse into each other. This is deliberate: a shrunken plot that hid its own labels would be worse than a scroll bar.

What matters for your reports is that copies and exports ignore the on-screen size entirely. Every graph you Copy or export is laid out fresh at full size, at least 1000 by 640 and rendered at triple resolution with the correct print resolution attached, so it drops into Word or PowerPoint sharp and at its natural scale. In short, a graph copied from a tiny window still looks right in the document. For comfortable viewing on screen, maximise the window or drag it larger.

Contacting support

If a message here does not match what you see, or a fix does not take, we are glad to help. The more you send with your first email, the faster we can answer.

Send this	Why it helps
A screenshot of the message	The exact wording tells us which check failed.
Your license key	Lets us find your license. It looks like OGDX-2026-XXXX-XXXX-XXXX.
The device name on the activation screen	Identifies the machine among your activations.
The app version and your operating system	The version is in Help › About OptiGein Design. It rules out version-specific behaviour.
The .ogdx file, for a data or analysis question	Lets us reproduce exactly what you are seeing.

Never send your activation key.
The activation key is the secret half of your license. Support never needs it, and you should not put it in an email or a screenshot. The license key on its own is enough for us to help.

Reach us by email at info@geinforce.com or by phone at +91 9960867592. You can also find us at geinforce.com. We are Geinforce Technology Private Limited, the makers of OptiGein Design.

Index

A

activation key 19, 20, 24
alias 38, 39, 40, 41, 51, 73, 122, 128, 132, 150
ANOVA 10, 30, 55, 59, 64, 66, 74, 75, 79, 83, 88, 90, 113, 115, 119, 122, 124, 129, 131, 133, 134, 135, 136, 137, 150
axial 36, 45, 47, 53, 108, 126, 127

B

Box-Behnken 47
Box-Cox 72, 80, 81, 83

C

central composite 36, 45, 47, 49, 60, 67, 69, 76, 82, 84, 93, 102, 108, 111, 118, 124
centre point 34, 36, 60, 67, 68, 75, 82, 84, 110, 111, 113, 118, 121, 150
coded 46, 76, 77, 78, 85, 87, 106, 107, 129, 132
confidence interval 76, 80, 117, 130
confirmation 19, 20, 97, 104, 105, 117, 119, 124, 130, 131, 151
contour 30, 84, 85, 86, 114, 129, 133
Cook's distance 82
correlation 65, 69

D

definitive screening 33, 49
design wizard 25, 27, 36, 138
desirability 31, 92, 93, 94, 95, 96, 97, 101, 116, 119, 150
DFFITS 82
diagnostics 10, 30, 72, 74, 79, 84, 90, 91, 102, 113, 115, 119, 122, 149, 150

E

equation 51, 77, 78, 107, 113, 129
evaluation 59, 62, 67, 70, 110, 124

F

factorial 9, 33, 34, 36, 37, 38, 44, 45, 47, 50, 69, 108, 120, 124, 125, 150
fit summary 30, 51, 55, 64, 66, 72, 79, 111, 119, 122, 128, 131, 135, 149, 150
fraction of design space 67, 70

G

graphical optimization 101, 117, 119

I

import 9, 25, 33, 37, 41, 55, 58, 59, 60, 77, 78, 92, 94, 95, 103, 106, 116, 120, 134, 138, 150
interaction 33, 41, 42, 43, 45, 48, 49, 53, 56, 67, 70, 72, 76, 84, 86, 88, 89, 111, 112, 114, 120, 122, 123, 124, 149, 150

L

lack of fit 48, 68, 73, 75, 110, 111, 113, 126, 150
lattice 51, 52, 53, 54, 125, 126
leverage 59, 68, 82, 110
license 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 139, 147, 148, 149

M

mixture 9, 34, 36, 37, 50, 51, 52, 53, 54, 55, 56, 90, 125, 126, 128, 129, 131
model graphs 30, 59, 74, 84, 114, 122, 123, 129, 135

N

numerical optimization 107, 119, 150

O

orthogonal 43, 69, 76

P

p-value 30, 73, 75, 79, 90, 106, 111, 113, 122
Plackett-Burman 40, 41
point prediction 51, 97, 102, 103, 104, 106, 117, 119, 130, 131
power 41, 69, 80, 91, 110, 134, 151, 152
prediction interval 105, 106, 117
PRESS 10, 14, 20, 21, 26, 50, 57, 58, 61, 65, 71, 72, 73, 76, 79, 95, 109, 111, 116, 121, 126, 147, 148, 149

R

replicate 34, 39, 40, 44, 45, 47, 68, 73, 75, 108, 121, 150
residual 10, 59, 68, 72, 75, 78, 79, 80, 82, 83, 113, 119
resolution 38, 40, 41, 42, 82, 120, 121, 133, 134, 152
response surface 9, 32, 33, 37, 44, 45, 49, 50, 67, 69, 70, 84, 85, 108, 124, 125, 129, 150
run sheet 10, 28, 37, 39, 47, 54, 57, 60, 61, 62, 63, 66, 67, 69, 82, 109, 121, 126, 127, 132, 133, 136, 137, 140, 149

S

screening 33, 37, 42, 45, 49, 121, 124, 150
simplex 50, 51, 52, 53, 125, 126, 131

T

Taguchi 43
tolerance interval 97, 107, 117

transform 64, 71, 72, 75, 80, 111, 132, 149, 150

V

VIF 69, 76