# 100 Challenging Numerical Problems

A reproducible benchmark and research challenge collection for Version 3.0.1

Panna Lal Patodia | Patodia Infotech Private Limited | 5 November 2026

## Scope and verification status

Problems 1–50 retain the executable mathematical models and published references from the supplied 50-problem benchmark volume. Their PASS statements refer to the original Windows regression reported in that document; they have not been rerun for this edition.

Problems 51–100 extend the collection into exact integer arithmetic, Diophantine systems, differentiation, tabular differentiation, interpolation, numerical limits, special functions and further quadrature. These are proposed additional regression cases adapted from your test archive, plus five explicitly marked new models. They are NOT claimed to have passed the current Windows build. Some source test models may be deliberately difficult or unsupported; validation is part of the exercise.

For reproducibility, the companion ZIP contains standalone UTF-8 text models for Problems 51–100. The first 50 problem statements are embedded below; their original test and verification notes are preserved. For any problem involving numerical methods, compare both residual and forward error when a trustworthy reference exists. Numerical stability and completeness are distinct properties.

## Coverage

1–10 Nonlinear equation systems

11–20 Quadrature

21–30 ODE and BVP

31–40 PDE

41–50 Integral equations

51–60 Exact integer and Diophantine problems

61–68 Multivariate and higher order differentiation

69–72 Tabular numerical differentiation

73–77 Interpolation and extrapolation

78–87 Finite and infinite limits

88–93 Compute special functions

94–100 Further difficult quadrature

## Part One Established 50 problem benchmark

## Nonlinear equation systems

### 1 01 Exp Polynomial 3D

```
x^4*exp(y-4)-2*x*y+y^2+3*x+z^3=20
x*y^3+y*exp(x-1)-z^2=64
```

```
z^3+x*y+sin(z-2)+log(1+(x-1)^2)=12
bound x=[-3,3]
bound y=[-1,6]
bound z=[-3,4]
```
Known planted solution(s): x=1, y=4, z=2

Additional reference branches (1): x=2.82480072, y=2.60913563, z=1.53447051

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 1/1; maximum independently reevaluated residual 7.1054274e-15.

## 2 02 Negative Coordinates 3D

```
exp(x+1)+y^3-z*x+sin(y-2)=12
x^3*y+z^2+cos(z-3)=8
x*y*z+(y-2)^4+log(1+(x+1)^2)=-6
bound x=[-4,4]
bound y=[-3,5]
bound z=[-3,6]
```
Known planted solution(s): x=-1, y=2, z=3

Additional reference branches (1): x=-2.61591352, y=0.776199382, z=4.6920667

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 1/1; maximum independently reevaluated residual 3.5527137e-15.

## 3 03 Coupled 4D

```
x^2*y+z^3*w+exp(w-3)=0
x*y*z+w^2+sin(x-2)=7
y^4+z^2*x-w+cos(y+1)=1
x^3+w*z-y^2+log(1+(z-1)^2)=10
bound x=[-4,5]
bound y=[-4,4]
bound z=[-4,5]
bound w=[-4,6]
```
Known planted solution(s): w=3, x=2, y=-1, z=1

Additional reference branches (1): w=3.05762893, x=2.34329536, y=0.912055386, z=-1.25662906

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 1/1; maximum independently reevaluated residual 1.7763568e-15.

## 4 04 Multibranch 4D

```
x^2*y+exp(z-3)+w^3=4
x*z+y^3-w^2+cos(x+2)=-5
z^2+x*y*w+sin(w+1)=11
x^3+y*z^2+w^4+log(1+(y-1)^2)=2
bound x=[-5,4]
bound y=[-4,5]
bound z=[-4,6]
bound w=[-5,5]
```
Known planted solution(s): w=-1, x=-2, y=1, z=3

Additional reference branches (6): w=-1.76606759, x=-2.4983143, y=1.44321304, z=2.30772049; w=0.604837429, x=-1.82827119, y=0.731550336, z=3.28778538; w=1.45072376, x=-1.10938339, y=-0.194030759, z=3.17025521; w=1.60187438, x=0.455159987, y=-0.54182807, z=-3.29865995; w=1.90975557, x=-1.30049527, y=-1.75555853, z=-2.53181352; w=2.6454757, x=-3.62661221, y=-1.10504859, z=-0.93854512

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 6/6; maximum independently reevaluated residual 5.3290705e-15.

## 5 05 Unbounded 5D

```
x*y*z+w^2+v^3+exp(x-1)=5
x^3+z^2*y+w*v+cos(z-2)=-9
y^2*w-z*v+x+sin(y+2)=15
x*z*w+y*v^2+w^2=13
v^4+x*y*w-z^3+log(1+(v+1)^2)=-13
```

Known planted solution(s): v=-1, w=3, x=1, y=-2, z=2

Additional reference branches (4): v=-3.71278548, w=-7.7358346, x=0.701486998, y=-1.05504148, z=5.95210657; v=-3.02801593, w=5.90989566, x=-0.984348326, y=0.510961471, z=4.574523; v=2.56763204, w=2.77057651, x=2.53324638, y=-2.81314673, z=3.40101941; v=3.98273981, w=2.29393592, x=5.01602085, y=-3.91022834, z=6.06293556

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 4/4; maximum independently reevaluated residual 2.4868996e-14.

The full benchmark for this case removes the optional search bounds to cover branches outside the original bounded box.

## 6 06 Unbounded 5D

```
x^2*z+y*w^3+exp(v-3)=5
x*y*v-z^3*w+sin(w-2)=8
z^2*v+y^4+x*w+cos(x-2)=9
v^2+x*z*w-y^3=4
x^3+y*z*v-w^2+log(1+(z+1)^2)=1
```

Known planted solution(s): v=3, w=2, x=2, y=1, z=-1

Additional reference branches (9): v=-16.132797, w=-3.93113976, x=7.42771192, y=4.68655942, z=5.25121891; v=-4.84437339, w=5.23141495, x=3.02348718, y=0.113501626, z=-1.23072318; v=-3.11398097, w=1.48433909, x=-1.25390744, y=1.93861775, z=-0.853691469; v=-0.087058664, w=-1.3263955, x=0.671848632, y=-1.77526529, z=1.79820439; v=0.199951143, w=-0.0812906438, x=-1.02263146, y=-1.5279872, z=4.72221237; v=1.39501343, w=-0.926928112, x=-1.58481467, y=1.07878824, z=2.25282042; v=1.42932653, w=-0.574146113, x=-1.38660509, y=0.385879634, z=2.53039708; v=2.69964132, w=-1.30276986, x=1.33695817, y=-0.375690247, z=1.91823836; v=2.9708019, w=1.76047506, x=1.92355117, y=1.28240933, z=-0.802231355

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 9/9; maximum independently reevaluated residual 5.4534155e-13.

The full benchmark for this case removes the optional search bounds to cover branches outside the original bounded box.

## 7 07 Symmetric 3D

```
x^2+y*z+exp(x^2-1)=8
x^2*y+z^3+sin(y-2)=29
z^2+y^3+cos(x^2-1)=18
bound x=[-4,4]
bound y=[-3,5]
bound z=[-3,6]
```

Known planted solution(s): x=-1, y=2, z=3; x=1, y=2, z=3

No additional reference roots required in this regression.

Windows regression: PASS; independent reference-run coverage: planted roots 2/2; reference coverage 0/0; maximum independently reevaluated residual 0.

## 8 08 Two Branches 4D

```
(x-1)*(x-2)+y^2+z*w+exp(y+1)=8
x^2-3*x+z^2*w+cos(z-2)=11
(x-1)*(x-2)*w+y*z+sin(w-3)=-2
z^3+w*y^2+log(1+(x-1)^2*(x-2)^2)=11
```

```
bound x=[-3,5]
bound y=[-4,4]
bound z=[-3,5]
bound w=[-3,6]
```
Known planted solution(s): w=3, x=1, y=-1, z=2; w=3, x=2, y=-1, z=2

Additional reference branches (2): w=5.54100392, x=0.993173363, y=1.64377949, z=-1.58367527; w=5.54100392, x=2.00682664, y=1.64377949, z=-1.58367527

Windows regression: PASS; independent reference-run coverage: planted roots 2/2; reference coverage 2/2; maximum independently reevaluated residual 1.7763568e-15.

## 9 09 Symmetric 5D

```
x^2*y+z^2+w*v+exp(x^2-1)=0
(x^2-1)*z+y^3+w^2+sin(v-3)=5
z^3+w*v+y^2+cos(x^2-1)=4
x^2*z*w+v^2-y^2=4
z*y*v+w^3+log(1+(x^2-1)^2)=-2
bound x=[-4,4]
bound y=[-4,5]
bound z=[-4,5]
bound w=[-5,4]
bound v=[-4,6]
```
Known planted solution(s): v=3, w=-2, x=-1, y=1, z=2; v=3, w=-2, x=1, y=1, z=2

Additional reference branches (2): v=-2.37603834, w=1.50493851, x=-0.467623012, y=1.48176683, z=1.6715186; v=-2.37603834, w=1.50493851, x=0.467623012, y=1.48176683, z=1.6715186

Windows regression: PASS; independent reference-run coverage: planted roots 2/2; reference coverage 2/2; maximum independently reevaluated residual 1.3322676e-15.

## 10 10 Ill Conditioned 5D

```
exp(x)+y^2*z+w*v+sin(y-1)=-4
x^3+z^2*w+v^2*y+cos(w-2)=7
y*z*w+v^3+exp(z+1)=-9
w^2*z+x*y+log(1+(v+2)^2)=-4
w^2*z+x*y+log(1+(v+2)^2)+0.000001*(v+2)=-4
bound x=[-4,4]
bound y=[-3,5]
bound z=[-5,4]
bound w=[-4,5]
bound v=[-5,4]
```
Known planted solution(s): v=-2, w=2, x=0, y=1, z=-1

No additional reference roots required in this regression.

Windows regression: PASS; independent reference-run coverage: planted roots 1/1; reference coverage 0/0; maximum independently reevaluated residual 0.

Sensitivity note: two equations differ by only $10^{-6}(v+2)$. Small raw residuals can admit an apparent continuum of nearly equivalent points in binary64. The verified planted root is (x,y,z,w,v)=(0,1,−1,2,−2); conditioning must be considered before interpreting nearby candidates as distinct roots.

Chapter 2 Numerical Quadrature

These ten cases include removable Taylor-remainder singularities, true improper endpoints, oscillatory cancellation and an infinite integration range. The reference values come from convergent series, beta and gamma identities, transformed integrals, or independent high-precision numerical evaluation.

# Numerical quadrature

## 11 Sixth-order exponential Taylor cancellation

```
quadrature
integrand = (exp(x)-1-x-x^2/2-x^3/6-x^4/24-x^5/120)/x^6
variable = x
interval = [0,1]
```

Challenge. The numerator vanishes to sixth order; direct floating-point subtraction loses almost all significant digits near zero.

Independent reference. Σ(j=0..∞) 1 / ((6+j)! (j+1)). Method: Absolutely convergent positive series.

Endpoint continuation or limit. 1/720.

Windows regression: PASS. Reference run accepted integral 0.00149711097650166021619640126326700194; independent absolute difference 1.230152782777290981441407932E-37.

## 12 Sixth-order alternating logarithmic remainder

```
quadrature
integrand = (log(1+x)-x+x^2/2-x^3/3+x^4/4-x^5/5)/x^6
variable = x
interval = [0,0.5]
```

Challenge. A sixth-order cancellation in the logarithm combines an alternating series with an endpoint 0/0 form.

Independent reference. Σ(j=0..∞) −(−1)^j (1/2)^(j+1) / ((6+j)(j+1)). Method: Alternating logarithmic series integrated termwise.

Endpoint continuation or limit. −1/6.

Windows regression: PASS. Reference run accepted integral -0.069403046847218254388219895398037639 8; independent absolute difference 5.091529871814204962414172127E-36.

## 13 Generalized binomial fourth-order remainder

```
quadrature
integrand = ((1+x)^(-1/2)-1+x/2-3*x^2/8+5*x^3/16)/x^4
variable = x
interval = [0,1]
```

Challenge. Severe binomial cancellation, with a finite value at x=0 that ordinary direct quadrature may reject.

Independent reference. 0.19535915775182208761702935992075671220350702202584905825801655806334 30. Method: Independent closed-form evaluation, checked with PARI/GP.

Endpoint continuation or limit. 35/128.

Windows regression: PASS. Reference run accepted integral 0.19535915775182208761702935992075669 4; independent absolute difference 1.820350702202584905825801656E-35.

## 14 Shifted and scaled seventh-order exponential remainder

```
quadrature
integrand = (exp(3*(x-1/4))-1-3*(x-1/4)-(3*(x-1/4))^2/2-(3*(x-1/4))^3/6-(3*(x-1/4))^4/24-(3*(x-1/4))^5/120-(3*
(x-1/4))^6/720)/(x-1/4)^7
variable = x
interval = [-1/4,3/4]
```

Challenge. The removable singularity is at the interior point x=1/4, not at an endpoint; the argument is both shifted and scaled.

Independent reference. 3^7 Σ(j=0..∞) (3/2)^(2j) / ((2j+1)(7+2j)!). Method: Symmetric-interval exponential series.

Endpoint continuation or limit. 3^7/7!.

Windows regression: PASS. Reference run accepted integral 0.438504710834567966674855254651948225; independent absolute difference 5.704562885785449925231164219E-35.

### 15 Fifty sine cycles against a semicircle endpoint

```
quadrature
integrand = sqrt(100^2*#pi^2-x^2)*sin(x)
variable = x
interval = [0,100*#pi]
```

Challenge. Large finite interval, 50 oscillations of sine, vanishing endpoint amplitude and extensive cancellation.

Independent reference. 298.435716494360384965604996063146023. Method: Published regression reference from independently checked Bessel-function identity.

Endpoint continuation or limit. 0 at x=100π.

Windows regression: PASS. Reference run accepted integral 298.435716494360384965604996063145826; independent absolute difference 1.97E-31.

### 16 An integrable cubic-root endpoint singularity

```
quadrature
integrand = x/sqrt(1-x^3)
variable = x
interval = [0,1]
```

Challenge. A true integrable right-endpoint singularity, not a removable singularity; substitution and endpoint treatment are essential.

Independent reference. Beta(2/3,1/2)/3. Method: Substitution $t=x^3$ gives a beta integral.

Endpoint continuation or limit. Divergent as x→1−.

Windows regression: PASS. Reference run accepted integral 0.86236985307659684498450500837701725 1; independent absolute difference 5.32948859528382801338587850 8E-32.

### 17 Logarithm multiplied by a strong algebraic singularity

```
quadrature
integrand = log(x)/x^(3/4)
variable = x
interval = [0,1]
```

Challenge. Unbounded logarithmic and power factors coexist at x=0, but the improper integral converges.

Independent reference. −16. Method: Differentiate $\int_0^1 x^{(a-1)}dx=1/a$ at a=1/4.

Endpoint continuation or limit. Divergent as x→0+.

Windows regression: PASS. Reference run accepted integral -16; independent absolute difference 0.

### 18 A quartic exponential on an infinite interval

```
quadrature
integrand = exp(-x^4)
variable = x
interval = [0,inf]
```

Challenge. Integration to infinity requires a valid tail estimate; the main contribution is concentrated near the origin.

Independent reference. Gamma(1/4)/4. Method: Substitution $t=x^4$ gives a gamma integral.

Endpoint continuation or limit. 0 as x→∞.

Windows regression: PASS. Reference run accepted integral 0.90640247705547707798267 1288966918022; independent absolute difference 2.12517484696919404569640 3097E-35.

### 19 A logarithmic remainder shifted to the right endpoint

```
quadrature
integrand = (log(1+(x-1)/2)-(x-1)/2+((x-1)/2)^2/2-((x-1)/2)^3/3)/(x-1)^4
variable = x
interval = [0,1]
```

Challenge. The removable singularity sits at the right endpoint x=1, after shifting and scaling the logarithm argument.

Independent reference. −Σ(j=0..∞) 1/(16·2^j·(j+1)·(j+4)). Method: Absolutely convergent shifted logarithm series.

Endpoint continuation or limit. −1/64 as x→1.

Windows regression: PASS. Reference run accepted integral -0.020054294558904840308862853463587389 2; independent absolute difference 6.66868016367147773103690552 8E-37.

### 20 Rapidly oscillating integral with an integrable endpoint singularity

```
quadrature
integrand = sin(50*x)/sqrt(1-x)
variable = x
interval = [0,1]
```

Challenge. Many sign changes combine with a genuine inverse-square-root singularity at the right endpoint; cancellation stresses error estimation.

Independent reference. $2\int_0^1 \sin(50(1-t^2))dt$. Method: Endpoint-flattening substitution $x=1-t^2$ and high-precision Gauss–Legendre.

Endpoint continuation or limit. Integrable divergence as x→1.

Windows regression: PASS. Reference run accepted integral -0.197546486506185767931863116792486012; independent absolute difference 1.253143982378766771558661925E-32.

Chapter 3 Ordinary Differential Equations and Boundary Value Problems

The ODE/BVP suite combines removable forcing singularities, regular singular initial points, stiff relaxation, fast oscillation, nonlinear boundary operators and mixed Robin conditions. In contrast with the earlier local development snapshot, all ten models passed on the tested Windows 3.0.1 executable.

## Ordinary differential equations and boundary value problems

### 21 Fourth-order binomial remainder as an initial-value forcing

```
ode
y'=((1+x)^(-1/2)-1+x/2-3*x^2/8+5*x^3/16)/x^4
initial x=0
initial y=0
interval x=[0,1]
rtol=1e-10
atol=1e-12
output points=11
```

Challenge. The ODE begins exactly at a fourth-order cancellation point; its source has a finite derivative despite the displayed 0/0 form.

Analytical reference. The exact terminal value is the independently checked binomial-remainder quadrature integral from Chapter 2, Problem 13. The derivative at zero is the removable limit 35/128.

Specified reference at y(1.0): 0.195359157751822087617029359920756712 2035.

Windows benchmark: PASS; independently checked absolute error 2.642102382970640079243287796e-12.

### 22 Sixth-order exponential remainder driving an ODE

```
ode
y'=(exp(x)-1-x-x^2/2-x^3/6-x^4/24-x^5/120)/x^6
initial x=0
```

```
initial y=0
interval x=[0,1]
rtol=1e-10
atol=1e-12
output points=11
```
Challenge. The source subtracts six nearly equal Taylor terms and tests stable evaluation at the initial point.

Analytical reference. Integrating the absolutely convergent remainder series gives Σ(j≥0)1/[(j+1)(j+6)!]; the source continuation at x=0 equals 1/720.

Specified reference at y(1.0): 0.00149711097650166021619640126326700206 30.

Windows benchmark: PASS; independently checked absolute error 3.09945161964012632670020 6302e-14.

## 23 Bessel equation at a regular singular initial point

```
ode
x*y''+y'+x*y=0
initial x=0
initial y=1
initial y'=0
interval x=[0,1]
rtol=1e-10
atol=1e-12
output points=11
```
Challenge. At x=0 the highest-derivative coefficient vanishes; the compatible regular solution must be continued by a local series, not arbitrary epsilon substitution.

Analytical reference. The regular solution is $J_0(x)=\Sigma(k\geq 0)(-1)^k(x^2/4)^k/(k!)^2$. In particular y'(0)=0 and y''(0)=−1/2.

Specified reference at y(1.0): 0.765197686557966551449717526102663 2209.

Windows benchmark: PASS; independently checked absolute error 1e-17.

## 24 Coupled state and exponential remainder in the Jacobian

```
ode
y'=y+(exp(t)-1-t)/t^2
initial t=0
initial y=0
interval t=[0,1]
rtol=1e-10
atol=1e-12
output points=11
```
Challenge. Taylor regularization is required inside a state-dependent differential equation and its derivative machinery.

Analytical reference. The exact solution is y(t)=e^t−(e^t−1)/t, interpreted by continuity at t=0. Differentiate to verify $y'=y+(e^t-1-t)/t^2$; y(1)=1.

Specified reference at y(1.0): 1.

Windows benchmark: PASS; independently checked absolute error 1.07376e-11.

## 25 High-frequency harmonic oscillator with one hundred radians

```
ode
y''=-10000*y
initial x=0
initial y=0
initial y'=100
interval x=[0,1]
rtol=1e-10
atol=1e-12
output points=101
```
Challenge. A high-frequency oscillator stresses adaptive step size, phase accuracy and dense output.

Analytical reference. The solution is y(x)=sin(100x), so the endpoint is sin(100). Accurate integration requires resolving the oscillations between output points.

Windows benchmark: PASS; independently checked absolute error 5.216101e-10.

### 26 A stiff relaxation problem with an exact rapidly decaying mode

```
ode
y'=-1000*(y-exp(-t))-exp(-t)
initial t=0
initial y=2
interval t=[0,1]
method=SDIRK2
rtol=1e-8
atol=1e-11
output points=101
```

Challenge. Two time scales differ by a factor of 1000; L-stable implicit integration should control fast-mode decay without oversolving the slow mode.

Analytical reference. The exact solution is y(t)=exp(−t)+exp(−1000t), obtained by subtracting exp(−t) and solving the homogeneous stiff mode.

Windows benchmark: PASS; independently checked absolute error 1.49932222e-9.

### 27 Boundary value problem with a stable exponential source

```
bvp
y''=(exp(x)-1-x-x^2/2)/x^3
interval x=[0,1]
boundary left y=0
boundary right y=0
rtol=1e-7
atol=1e-9
output points=101
```

Challenge. A second-order BVP evaluates a removable source singularity at the boundary, and the forcing derivative affects the BVP Jacobian.

Analytical reference. If $R_3(x)=\Sigma(j\geq 0)x^j/(j+3)!$, then $y(x)=\Sigma(j\geq 0)x^{(j+2)}/[(j+1)(j+2)(j+3)!]-Cx$, where C is the same series evaluated at x=1. Both boundary conditions hold.

Windows benchmark: PASS; independently checked absolute error 1.554459089472194045736858455e-14.

### 28 Sharp exponential boundary layer in a linear BVP

```
bvp
y''=10000*y
interval x=[0,1]
boundary left y=1
boundary right y=0
rtol=1e-6
atol=1e-8
max nodes=4000
output points=101
```

Challenge. A layer of width approximately 0.01 lies next to the left boundary; coarse grids can have very small algebraic residuals but large continuum errors.

Analytical reference. The unique solution is y(x)=sinh(100(1−x))/sinh(100); at x=0.01 the value is close to e^(−1).

Windows benchmark: PASS; independently checked absolute error 4.6578446e-10.

### 29 Nonlinear cubic BVP with a manufactured sine solution

```
bvp
y''=y^3-pi^2*sin(pi*x)-sin(pi*x)^3
interval x=[0,1]
boundary left y=0
boundary right y=0
guess y=sin(pi*x)
rtol=1e-7
```

```
atol=1e-9
output points=101
```
Challenge. The nonlinear residual and Jacobian must preserve an exact solution under a stiff cubic reaction term.

Analytical reference. Insert $y(x)=\sin(\pi x)$: $y''=-\pi^2\sin(\pi x)$, and the cubic $y^3$ cancels the manufactured $-\sin^3(\pi x)$ source.

Specified reference at y(0.5): 1.

Windows benchmark: PASS; independently checked absolute error 3.573142e-11.

### 30 Two Robin conditions with an exponentially growing exact solution

```
bvp
y''=4*y
interval x=[0,1]
boundary left y+y'=3
boundary right y-y'=-exp(2)
rtol=1e-8
atol=1e-10
output points=101
```
Challenge. Both boundaries involve derivatives, so boundary residuals and endpoint Jacobians must be evaluated accurately.

Analytical reference. For y(x)=e^(2x), y"=4y, y(0)+y'(0)=3, and $y(1)-y'(1)=-e^2$. These two Robin conditions determine the constants.

Windows benchmark: PASS; independently checked absolute error 2.69052e-11.



The PDE suite tests Poisson, Laplace, heat and wave equation formulations under challenging source and boundary expressions. The models have manufactured exact fields and independent nodal-error checks. A fixed-grid PASS is not a guarantee of continuum accuracy.

## Partial differential equations

### 31 Poisson equation with a thin exponential boundary layer

```
pde
type=poisson
u_xx+u_yy=(400-pi^2)*exp(-20*x)*sin(pi*y)
domain x=[0,1]
domain y=[0,1]
boundary left u=sin(pi*y)
boundary right u=exp(-20)*sin(pi*y)
boundary bottom u=0
boundary top u=0
grid x=129
grid y=129
refine=false
exact solution=exp(-20*x)*sin(pi*y)
```
Challenge. Boundary layer scale 1/20; source and second derivatives vary by orders of magnitude across the unit square. Test error against Δx and compare residual to true nodal error.

Exact reference: u=exp(-20*x)*sin(pi*y).

Analytical verification. For $u=e^{(-20x)}\sin(\pi y)$, $u_{xx}=400u$ and $u_{yy}=-\pi^2u$. The boundary data match the exact function.

Windows benchmark: PASS, maximum independent nodal error 0.0012418059948708093; acceptance threshold 0.003; 16641 nodes.

Grid refinement ratio E(coarse)/E(fine)=3.98501.

## 32 Exponential harmonic field with rapid transverse oscillation

```
pde
type=laplace
domain x=[0,1]
domain y=[0,1]
boundary left u=exp(-12)*cos(12*y)
boundary right u=cos(12*y)
boundary bottom u=exp(12*(x-1))
boundary top u=exp(12*(x-1))*cos(12)
grid x=129
grid y=129
refine=false
exact solution=exp(12*(x-1))*cos(12*y)
```

Challenge. An exponential growth ratio e^12 occurs across x, accompanied by almost two full cosine cycles in y. Accurate absolute errors require resolving the near-right-edge structure.

Exact reference: u=exp(12*(x-1))*cos(12*y).

Analytical verification. The exact field e^(12(x−1))cos(12y) is harmonic because u_xx=144u and u_yy=−144u. All four boundary traces are consistent at corners.

Windows benchmark: PASS, maximum independent nodal error 0.00029705488922521228; acceptance threshold 0.001; 16641 nodes.

Grid refinement ratio E(coarse)/E(fine)=3.986219.

## 33 Pure Neumann Poisson equation with high spatial frequencies

```
pde
type=poisson
u_xx+u_yy=-64*pi^2*cos(8*pi*x)-18*pi^2*cos(6*pi*y)
domain x=[0,1]
domain y=[0,1]
boundary left normal=0
boundary right normal=0
boundary bottom normal=0
boundary top normal=0
mean value=0
grid x=129
grid y=129
refine=false
exact solution=cos(8*pi*x)+0.5*cos(6*pi*y)
```

Challenge. High-frequency forcing combined with a singular Neumann operator; the zero-mean constraint removes the constant null mode. Verify flux compatibility separately.

Exact reference: u=cos(8*pi*x)+0.5*cos(6*pi*y).

Analytical verification. The Laplacian gives $-(8\pi)^2\cos(8\pi x)-0.5(6\pi)^2\cos(6\pi y)$. Both outward normal fluxes vanish, and the domain mean is zero.

Windows benchmark: PASS, maximum independent nodal error 0.0041235343330889851; acceptance threshold 0.01; 16641 nodes.

Grid refinement ratio E(coarse)/E(fine)=4.021025.

## 34 Harmonic field with a Taylor-dependent Robin boundary coefficient

```
pde
type=laplace
domain x=[0,1]
domain y=[0,1]
boundary left robin=[1+((exp(y)-1-y-y^2/2)/y^3)^2,1,((exp(y)-1-y-y^2/2)/y^3)^2*cos(y)]
boundary right u=exp(1)*cos(y)
boundary bottom u=exp(x)
boundary top u=exp(x)*cos(1)
grid x=65
grid y=65
```

```
refine=false
exact solution=exp(x)*cos(y)
```
Challenge. A variable Robin impedance contains a removable cubic exponential remainder at y=0. Its correct limit is 1/6; boundary residual and Jacobian must use the stable continuation.

Exact reference: u=exp(x)*cos(y).

Analytical verification. The harmonic reference u=e^x cos y satisfies u_xx+u_yy=0. At x=0, outward normal derivative is −cos y. With α=1+R(y)$^2$, β=1, the Robin right side is α cos y−cos y=R(y)$^2$cos y.

Windows benchmark: PASS, maximum independent nodal error 1.3700884653355949e-05; acceptance threshold 0.0001; 4225 nodes.

Grid refinement ratio E(coarse)/E(fine)=3.99796.

## 35 Oscillatory Poisson source with an interior Taylor continuation

```
pde
type=poisson
u_xx+u_yy=-25*pi^2*sin(3*pi*x)*sin(4*pi*y)+((exp(2*x-1)-1-(2*x-1)-(2*x-1)^2/2)/(2*x-1)^3)-((exp(2*x-1)-1-(2*x-1)-(2*x-1)^2/2)/(2*x-1)^3)
domain x=[0,1]
domain y=[0,1]
boundary left u=0
boundary right u=0
boundary bottom u=0
boundary top u=0
grid x=129
grid y=129
refine=false
exact solution=sin(3*pi*x)*sin(4*pi*y)
```
Challenge. The numerical forcing is rapidly oscillatory and contains a removable interior quotient. The cancellation is intentional, testing the expression evaluator independently of the finite-difference truncation.

Exact reference: u=sin(3*pi*x)*sin(4*pi*y).

Analytical verification. The reference u=sin(3πx)sin(4πy) has Δu=−25π$^2$u. Equal positive and negative Taylor-remainder terms cancel analytically; their individual continuation at x=1/2 is 1/6.

Windows benchmark: PASS, maximum independent nodal error 0.0006769514064968174; acceptance threshold 0.002; 16641 nodes.

Grid refinement ratio E(coarse)/E(fine)=4.004675.

## 36 Variable-diffusivity heat equation with a rapidly decaying spatial mode

```
pde
type=heat
u_t=(1+x)*u_xx+((1+x)*100*pi^2-100)*exp(-100*t)*sin(10*pi*x)
domain x=[0,1]
interval t=[0,0.1]
initial u=sin(10*pi*x)
boundary left u=0
boundary right u=0
grid x=129
grid t=201
refine=false
exact solution=exp(-100*t)*sin(10*pi*x)
```
Challenge. Diffusivity varies by a factor of two, with ten half-waves and strong temporal decay. Large forcing and diffusion terms must cancel accurately.

Exact reference: u=exp(-100*t)*sin(10*pi*x).

Analytical verification. For u=e^(−100t)sin(10πx), u_t=−100u and u_xx=−100π$^2$u. Substitution gives the manufactured source [(1+x)100π$^2$−100]u.

Windows benchmark: PASS, maximum independent nodal error 0.0042976151038750521; acceptance threshold 0.01; 25929 nodes.

## 37 Moving Gaussian interior layer in convection reaction diffusion

```
pde
type=heat
u_t=u_xx+u_x+u+(198+30*(10*(x-0.3-0.5*t))-400*(10*(x-0.3-0.5*t))^2)*exp(-(10*(x-0.3-0.5*t))^2-t)
domain x=[0,1]
interval t=[0,0.2]
initial u=exp(-(10*(x-0.3))^2)
boundary left u=exp(-(10*(0-0.3-0.5*t))^2-t)
boundary right u=exp(-(10*(1-0.3-0.5*t))^2-t)
grid x=129
grid t=201
refine=false
exact solution=exp(-(10*(x-0.3-0.5*t))^2-t)
```

Challenge. A narrow moving interior feature, nonzero time-dependent boundary values and large derivatives stress advection/reaction source handling and grid resolution.

Exact reference: u=exp(-(10*(x-0.3-0.5*t))^2-t).

Analytical verification. Set z=10(x−0.3−0.5t), u=e^(−$z^2$−t). Then u_t=(−1+10z)u, u_x=−20zu, u_xx=(400$z^2$−200)u. Therefore u_t−u_xx−u_x−u=(198+30z−400$z^2$)u.

Windows benchmark: PASS, maximum independent nodal error 0.0009711563177906557; acceptance threshold 0.003; 25929 nodes.

## 38 Two-scale heat diffusion with a rapidly extinguished high mode

```
pde
type=heat
u_t=u_xx
domain x=[0,1]
interval t=[0,0.05]
initial u=sin(pi*x)+0.01*sin(20*pi*x)
boundary left u=0
boundary right u=0
grid x=129
grid t=201
refine=false
exact solution=exp(-pi^2*t)*sin(pi*x)+0.01*exp(-400*pi^2*t)*sin(20*pi*x)
```

Challenge. The high mode is small in amplitude but extremely fast in time; check short-time phase-free decay and spatial resolution without masking error under the slow mode.

Exact reference: u=exp(-pi^2*t)*sin(pi*x)+0.01*exp(-400*pi^2*t)*sin(20*pi*x).

Analytical verification. Each sine eigenmode independently decays as exp(−(kπ)$^2$t). The k=20 mode has a decay rate 400 times that of k=1.

Windows benchmark: PASS, maximum independent nodal error 0.00081737490161648552; acceptance threshold 0.003; 25929 nodes.

## 39 High-frequency wave propagation over many temporal oscillations

```
pde
type=wave
u_tt=4*u_xx
domain x=[0,1]
interval t=[0,0.2]
initial u=sin(15*pi*x)
initial u_t=0
boundary left u=0
boundary right u=0
grid x=241
grid t=201
refine=false
exact solution=sin(15*pi*x)*cos(30*pi*t)
```

Challenge. Fifteen spatial half-waves with a speed-two hyperbolic operator; phase dispersion accumulates during almost three temporal cycles and CFL restrictions matter.

Exact reference: u=sin(15*pi*x)*cos(30*pi*t).

Analytical verification. The standing-wave reference sin(15πx)cos(30πt) satisfies u_tt=−900π²u and 4u_xx=−900π²u. Initial displacement and zero initial velocity are compatible.

Windows benchmark: PASS, maximum independent nodal error 0.021420452300560271; acceptance threshold 0.03; 48441 nodes.

Grid refinement ratio E(coarse)/E(fine)=4.009649.

### 40 Taylor-remainder-dependent diffusion with a nontrivial exact sine state

```
pde
type=heat
u_t=(2+((exp(x)-1-x-x^2/2)/x^3)^2)*u_xx+pi^2*(1+((exp(x)-1-x-x^2/2)/x^3)^2)*exp(-pi^2*t)*sin(pi*x)
domain x=[0,1]
interval t=[0,0.2]
initial u=sin(pi*x)
boundary left u=0
boundary right u=0
grid x=65
grid t=121
refine=false
exact solution=exp(-pi^2*t)*sin(pi*x)
```

Challenge. Nonconstant positive diffusivity and nontrivial spatial curvature require stable evaluation of a high-order removable quotient in both the coefficient and the forcing.

Exact reference: u=exp(-pi^2*t)*sin(pi*x).

Analytical verification. Write R(x)=[exp(x)−1−x−x²/2]/x³, with R(0)=1/6. For u=e^(−π²t)sin(πx), u_t=−π²u and u_xx=−π²u. Thus a(x)=2+R² and source π²(1+R²)u exactly balance.

Windows benchmark: PASS, maximum independent nodal error 0.00011597456909051562; acceptance threshold 0.001; 7865 nodes.

Chapter 5 Integral Equations

The integral-equation chapter combines causal Volterra memory, scalar near-resonant Fredholm resolvents, oscillatory and exponential kernels, two-component Volterra systems, nonlinear Volterra Hammerstein forms and interior Taylor-remainder continuation. Coupled Fredholm and nonlinear Fredholm models are outside the tested engine grammar and are not claimed here.

## Integral equations

### 41 Volterra equation with exponential growth and oscillatory memory

```
integral
type=volterra
interval x=[0,1]
forcing=(exp(x)+cos(x)-sin(x))/2
kernel=cos(x-s)
exact solution=exp(x)
grid x=257
refine=false
```

Challenge. Nonseparable oscillatory memory coupled to an exponentially increasing solution; independently integrate the convolution.

Exact reference: u(x)=exp(x).

Analytical verification. For u(s)=exp(s), the convolution integral $\int_0^x \cos(x-s)\exp(s)ds=[\exp(x)-\cos(x)+\sin(x)]/2$.

Windows benchmark: PASS, maximum independent nodal error 2.3524897709847892e-06; acceptance threshold 0.0002.

Grid refinement ratio E(coarse)/E(fine)=4.00007.

## 42 Volterra equation with cubic historical memory

```
integral
type=volterra
interval x=[0,1]
forcing=1+x+x^2-x^4/4-x^5/20-x^6/60
kernel=(x-s)^3
exact solution=1+x+x^2
grid x=129
refine=false
```

Challenge. A fourth-order accumulating memory operator tests long-range quadrature error and polynomial exactness.

Exact reference: u(x)=1+x+x^2.

Analytical verification. For u(s)=1+s+$s^2$, the cubic-memory integral is $x^4/4+x^5/20+x^6/60$.

Windows benchmark: PASS, maximum independent nodal error 1.0390851315378313e-05; acceptance threshold 0.0002.

Grid refinement ratio E(coarse)/E(fine)=4.00008.

## 43 Near-resonant Fredholm equation with an almost-singular operator

```
integral
type=fredholm
interval x=[0,1]
lambda=2.999
forcing=x*(1-2.999/3)
kernel=x*s
exact solution=x
grid x=257
refine=false
```

Challenge. Rank-one integral operator with eigenvalue 1/3 and coupling 2.999, close to the singular value 3; perturbations are amplified.

Exact reference: u(x)=x.

Analytical verification. The rank-one kernel xs has nonzero eigenvalue $\int_0^1 s^2 ds=1/3$. The coupling λ=2.999 is close to the singular value 3; the exact solution u=x has forcing x(1−λ/3).

Windows benchmark: PASS, maximum independent nodal error 1.2948531136203201e-12; acceptance threshold 0.003.

## 44 Fredholm equation with rapidly oscillatory nonseparable kernel

```
integral
type=fredholm
interval x=[0,1]
lambda=0.3
forcing=1-0.3*(sin(12*x)+sin(12*(1-x)))/12
kernel=cos(12*(x-s))
exact solution=1
grid x=257
refine=false
```

Challenge. An oscillatory kernel with twelve-radian phase changes tests cancellation and kernel integration accuracy.

Exact reference: u(x)=1.

Analytical verification. For u(s)=1, $\int_0^1 \cos(12(x-s))ds=[\sin(12x)+\sin(12(1-x))]/12$.

Windows benchmark: PASS, maximum independent nodal error 2.6645352591003757e-15; acceptance threshold 0.0001.

## 45 Fredholm equation with exponentially scaled rank-one kernel

```
integral
type=fredholm
interval x=[0,1]
lambda=0.1
```

```
forcing=exp(x)*(1-0.1*(exp(2)-1)/2)
kernel=exp(x+s)
exact solution=exp(x)
grid x=129
refine=false
```
Challenge. Large exponential variation in both arguments and a nontrivial resolvent eigenmode.

Exact reference: u(x)=exp(x).

Analytical verification. With u(s)=exp(s), the integral of exp(x+s)u(s) is exp(x)[exp(2)−1]/2.

Windows benchmark: PASS, maximum independent nodal error 2.6645352591003757e-15; acceptance threshold 0.0002.

## 46 Two-component Volterra system with asymmetric cross-memory

```
integral
type=volterra
components=2
interval x=[0,1]
forcing 1=1+x-0.2*(x+x^2/2)-0.1*(x^2-x^3/3)
forcing 2=2-x+0.15*x*(x+x^2/2)-0.1*(2*x-x^2/2)
kernel 1 1=0.2
kernel 1 2=0.1*s
kernel 2 1=-0.15*x
kernel 2 2=0.1
exact 1=1+x
exact 2=2-x
grid x=129
refine=false
```
Challenge. Asymmetric off-diagonal kernels and different solution gradients test multicomponent assembly.

Exact reference components: u1(x)=1+x, u2(x)=2-x.

Analytical verification. Substitute $u_1=1+x$ and $u_2=2-x$, then integrate each of the four polynomial component kernels over $0\le s\le x$.

Windows benchmark: PASS, maximum independent nodal error 1.1244089888062803e-06; acceptance threshold 0.0003.

Grid refinement ratio E(coarse)/E(fine)=4.

## 47 Two-component Volterra system with nonseparable cross-kernels

```
integral
type=volterra
components=2
interval x=[0,1]
lambda=0.2
forcing 1=1+x-0.2*(x^3/2+x^4/3+sin(x)*(2*x-x^2/2))
forcing 2=2-x-0.2*(cos(x)*(x^2/2+x^3/3)+3*x^2-5*x^3/6)
kernel 1 1=x*s
kernel 1 2=sin(x)
kernel 2 1=cos(x)*s
kernel 2 2=x+s
exact 1=1+x
exact 2=2-x
grid x=129
refine=false
```
Challenge. Full two-by-two integral coupling, trigonometric cross-kernels and independent manufactured forcings.

Exact reference components: u1(x)=1+x, u2(x)=2-x.

Analytical verification. For $u_1=1+x$ and $u_2=2-x$, the four causal terms are $x^3/2+x^4/3$, $\sin(x)(2x-x^2/2)$, $\cos(x)(x^2/2+x^3/3)$, and $3x^2-5x^3/6$.

Windows benchmark: PASS, maximum independent nodal error 2.0973438683569157e-06; acceptance threshold 0.0003.

Grid refinement ratio E(coarse)/E(fine)=4.00002.

## 48 Nonlinear Volterra Hammerstein equation with quadratic response

```
integral
type=volterra
interval x=[0,1]
lambda=0.1
forcing=1-0.2*x
kernel=1+u(s)^2
exact solution=1
grid x=129
refine=false
```

Challenge. Nonlinear unknown-dependent kernel requires a consistent diagonal Newton derivative and causal quadrature.

Exact reference: u(x)=1.

Analytical verification. At u=1, the nonlinear kernel $1+u(s)^2$ is 2, giving the manufactured right side 1−0.2x.

Windows benchmark: PASS, maximum independent nodal error 0; acceptance threshold 0.0004.

## 49 Nonlinear Volterra equation with exponential memory and global feedback

```
integral
type=volterra
interval x=[0,1]
lambda=0.1
forcing=1-0.1*2*(exp(x)-1)
kernel=exp(x-s)*(1+u(s)^2)
exact solution=1
grid x=129
refine=false
```

Challenge. A nonlinear exponentially weighted history couples each time point to the entire earlier interval.

Exact reference: u(x)=1.

Analytical verification. At u=1, the nonlinear kernel equals 2exp(x−s), whose integral is 2[exp(x)−1].

Windows benchmark: PASS, maximum independent nodal error 2.2199317391002893e-06; acceptance threshold 0.0005.

Grid refinement ratio E(coarse)/E(fine)=3.99998.

## 50 Near-resonant Fredholm equation with fifth-order shifted Taylor remainders

```
integral
type=fredholm
interval x=[0,1]
lambda=2.9999
forcing=x*(1-2.9999/3)+((exp((x-0.5))-1-(x-0.5)-(x-0.5)^2/2-(x-0.5)^3/6-(x-0.5)^4/24)/(x-0.5)^5)-((exp((x-0.5))-1-(x-0.5)-(x-0.5)^2/2-(x-0.5)^3/6-(x-0.5)^4/24)/(x-0.5)^5)
kernel=x*s+((exp((s-0.5))-1-(s-0.5)-(s-0.5)^2/2-(s-0.5)^3/6-(s-0.5)^4/24)/(s-0.5)^5)-((exp((s-0.5))-1-(s-0.5)-(s-0.5)^2/2-(s-0.5)^3/6-(s-0.5)^4/24)/(s-0.5)^5)
exact solution=x
grid x=257
refine=false
```

Challenge. Near-singularity of the Fredholm resolvent combines with severe cancellation at the interior collocation point s=0.5.

Exact reference: u(x)=x.

Analytical verification. The shifted fifth-order exponential remainder has continuation 1/120 at its pole. Its two signed copies cancel in the forcing and kernel. The remaining rank-one operator has λ=2.9999, close to the resonance λ=3.

Windows benchmark: PASS, maximum independent nodal error 1.2509326907661489e-11; acceptance threshold 0.003.

# Part Two  Additional 50 research challenges

Each executable model is included in the companion ZIP. Expected mathematical properties are listed where they can be stated without claiming an unperformed solver run. Source models are preserved as found, with comments and runtime settings from the test archive excluded.

# Exact integer and Diophantine problems

### 51  Bhaskar01

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
/*
A person spends exactly $100 to buy exactly 100 fruits, purchasing at least one bundle of each variety.
Bananas are sold in bundles of five for $3, oranges in bundles of seven for $5,
apples in bundles of nine for $7, and mangoes individually for $3 each.
Find every possible purchase.

Let x, y, z and w count purchases of the four offers.
Each variable is a positive integer because fractional bundles and the omission of a variety are excluded.
*/

var x,y,z,w
pos int x,y,z,w

3*x + 5*y + 7*z + 3*w = 100
5*x + 7*y + 9*z + w = 100

bound x = [1,100]
bound y = [1,100]
bound z = [1,100]
bound w = [1,100]
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 051_bhaskar01.txt | Status: new challenge; not rerun for this edition.

### 52  Bhaskar02

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
/*
Triangle problem attributed to Bhaskaracharya

Two sides of a triangle have lengths sqrt(13) and sqrt(5).
Its area is 4 square units.
Find all possible lengths of the third side.
*/

var a, b, c,s  // Order variables in this order while printing

a = sqrt(13)
b = sqrt(5)
s = (a+b+c)/2

sqrt(s*(s-a)*(s-b)*(s-c)) = 4

a+b >= c
a+c >= b
b+c >= a
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 052_bhaskar02.txt | Status: new challenge; not rerun for this edition.

## 53 Bhaskar03

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
/*
Triangle problem attributed to Bhaskaracharya

Two sides of a triangle have lengths sqrt(10)-sqrt(5)
and sqrt(6). Its base has length sqrt(18)-1.
Find the perpendicular height corresponding to this base.
*/

var a,b,c,s,h

a = sqrt(10)-sqrt(5)
b = sqrt(6)
c = sqrt(18)-1

s = (a+b+c)/2

s*(s-a)*(s-b)*(s-c) = (c*h/2)^2

h >= 0
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 053_bhaskar03.txt | Status: new challenge; not rerun for this edition.

## 54 Bhaskar04

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
pos int x, y

x % 5 = 1
y % 6 = 2
(y - x) % 3 = 2
(x + y) % 9 = 5
(x*y) % 7 = 6
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 054_bhaskar04.txt | Status: new challenge; not rerun for this edition.

## 55 Bhaskar05

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
var x, y, a
pos int x, a, y

5x - 13y = a
x + y = 30
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 055_bhaskar05.txt | Status: new challenge; not rerun for this edition.

## 56 Factors

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
pos int a, b
a * b = 12345678905
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 056_factors.txt | Status: new challenge; not rerun for this edition.

### 57 Factors02

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
pos int a, b
a * b = 123456789051
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 057_factors02.txt | Status: new challenge; not rerun for this edition.

### 58 Factors03

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
pos int a, b
a*b = 9444732965739290427393
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 058_factors03.txt | Status: new challenge; not rerun for this edition.

### 59 Sum Of Squares

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
pos int x,y
x^2 + 61*y^2=3567754103610864591238571O

bound x=[1,1e30]
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 059_sum_of_squares.txt | Status: new challenge; not rerun for this edition.

### 60 Sum Of Squares02

Research challenge. Large discrete search, arithmetic structure and complete-solution reporting.

```
pos int x,y
7x^2 +  11*y^2 = 22855060818840362

bound x=[1,1e30]
```

Verification target. Check integrality, exact substitution, gcd/feasibility, bounds, and whether the complete solution family is reported.

Executable file: 060_sum_of_squares02.txt | Status: new challenge; not rerun for this edition.

## Multivariate and higher order differentiation

### 61 Diff01

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = x^2*exp(y)+sin(x*z)+y^3*z

at x = 1
at y = 0
at z = 2

grad = true
hess = true
```

Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 061_diff01.txt | Status: new challenge; not rerun for this edition.

## 62 Diff02

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = x^3 + 2*x^2 - 5*x + 7
at x = 2
wrt = x
order = 1
```

Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 062_diff02.txt | Status: new challenge; not rerun for this edition.

## 63 Diff03

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = x^4-3*x^3+2*x^2-7*x+5
at x = 2
wrt = x
order = 2
```

Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 063_diff03.txt | Status: new challenge; not rerun for this edition.

## 64 Diff04

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = x^3*y^2+sin(x*y)

at x = 1
at y = 2

wrt = [x,y]
```

Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 064_diff04.txt | Status: new challenge; not rerun for this edition.

## 65 Diff05

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = exp(2*x)

at x = 0

wrt = x
order = 4
```

Verification target. Exact reference: $d^4\exp(2x)/dx^4$ at x=0 equals 16.

Executable file: 065_diff05.txt | Status: new challenge; not rerun for this edition.

## 66 Diff06

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = (exp(1-x^2)-1-(1-x^2)-(1-x^2)^2/2)/(1-x^2)^3
```

```
at x = 1

wrt = x
order = 1
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 066_diff06.txt | Status: new challenge; not rerun for this edition.

### 67 Diff07

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = sinh(2x)

at x = 0.1

wrt = x
order = 2
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 067_diff07.txt | Status: new challenge; not rerun for this edition.

### 68 Diff08

Research challenge. Derivative accuracy, higher orders and parameter coupling.

```
diff
expr = acosh(x)
at x = 2
wrt = x
order = 1
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 068_diff08.txt | Status: new challenge; not rerun for this edition.

## Tabular numerical differentiation

### 69 Diff Table01

Research challenge. Approximate derivatives from sampled and potentially noisy data.

```
diff
data = [
    [0, 0],
    [1, 1],
    [2, 4],
    [3, 9],
    [4, 16],
    [5, 25]
]

at x = [1.5, 2.5, 3.5]
order = 1
method = auto
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 069_diff-table01.txt | Status: new challenge; not rerun for this edition.

### 70 Diff Table03

Research challenge. Approximate derivatives from sampled and potentially noisy data.

```
diff
data = [[0,0],[1,1],[2,16],[3,81],[4,256],[5,625],[6,1296]]
order = 2
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 070_diff-table03.txt | Status: new challenge; not rerun for this edition.

### 71 Diff Table04

Research challenge. Approximate derivatives from sampled and potentially noisy data.

```
diff
data = [
    [0, 0],
    [0.05, 0.10017],
    [0.10, 0.20134],
    [0.15, 0.30452],
    [0.20, 0.41075],
    [0.25, 0.52110]
]

order = 1
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 071_diff-table04.txt | Status: new challenge; not rerun for this edition.

### 72 Diff Table05

Research challenge. Approximate derivatives from sampled and potentially noisy data.

```
diff
data = [
    [0.96, 0.7825361],
    [0.98, 0.7739332],
    [1.00, 0.7651977],
    [1.02, 0.7563321],
    [1.04, 0.7473390]
]

order = 1
```
Verification target. Check derivative order, variable selection, endpoint stencils, and compare with analytic or high-precision independent differences.

Executable file: 072_diff-table05.txt | Status: new challenge; not rerun for this edition.

## Interpolation and extrapolation

### 73 Interpolate01

Research challenge. Recover stable values at unsampled locations and check extrapolation.

```
interpolate
data = [[0,0],[1,1],[2,4],[3,9],[4,16],[5,25]]
at x = [1.5, 2.5, 3.5]
method = auto
```
Verification target. Exact reference: $y=x^2$; interpolated values 2.25, 6.25 and 12.25.

Executable file: 073_interpolate01.txt | Status: new challenge; not rerun for this edition.

## 74 Interpolate02

Research challenge. Recover stable values at unsampled locations and check extrapolation.

```
interpolate
data = [
    [0, 0],
    [1, 1]
    [2, 1.41421356237],
    [3, 1.73205080757],
    [4, 2],
    [5, 2.23606797750]
]

at x = [1.5, 2.5, 3.5, 6]

method = auto
degree = 4
extrapolate = true
```

Verification target. Compare against independently constructed interpolants; distinguish interpolation from extrapolation.

Executable file: 074_interpolate02.txt | Status: new challenge; not rerun for this edition.

## 75 Interpolate03

Research challenge. Recover stable values at unsampled locations and check extrapolation.

```
interpolate
data = [
    [0, 0],
    [1, 1]
    [2, 1.41421356237],
    [3, 1.73205080757],
    [4, 2],
    [5, 2.23606797750]
]

at x = [1.5, 2.5, 3.5]

method = cubic_spline
```

Verification target. Compare against independently constructed interpolants; distinguish interpolation from extrapolation.

Executable file: 075_interpolate03.txt | Status: new challenge; not rerun for this edition.

## 76 Interpolate Custom04

Research challenge. Recover stable values at unsampled locations and check extrapolation.

```
interpolate
data = [[-2,4],[-1,1],[0,0],[1,1],[2,4],[3,9]]
at x = [-1.5,0.5,2.5]
method = auto
```

Verification target. Exact reference: $y=x^2$, so values 2.25, 0.25 and 6.25.

Executable file: 076_interpolate_custom04.txt | Status: new challenge; not rerun for this edition.

## 77 Interpolate Custom05

Research challenge. Recover stable values at unsampled locations and check extrapolation.

```
interpolate
data = [[0,1],[1,2],[2,5],[3,10],[4,17],[5,26]]
at x = [0.25,1.75,4.5]
method = auto
```

Verification target. Exact reference: $y=x^2+1$, so values 1.0625, 4.0625 and 21.25.

Executable file: 077_interpolate_custom05.txt | Status: new challenge; not rerun for this edition.

# Finite and infinite limits

### 78 Limit04

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit
expr = (x^2 - 3x + 2) / (x - 1)

at x = 1
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 078_limit04.txt | Status: new challenge; not rerun for this edition.

### 79 Limit08

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit
expr = acot(x)

at x = 0
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 079_limit08.txt | Status: new challenge; not rerun for this edition.

### 80 Limit12

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit
expr = x - sqrt(x^2 + x)

at x = inf
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 080_limit12.txt | Status: new challenge; not rerun for this edition.

### 81 Limit15

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit
expr = (2 * exp(x) - 2) / x

at x = 0
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 081_limit15.txt | Status: new challenge; not rerun for this edition.

### 82 Limit20

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit
expr = abs(x)^sin(x)

at x = 0
```

Verification target. Exact reference: abs(x)^sin(x) tends to 1 at x=0.

Executable file: 082_limit20.txt | Status: new challenge; not rerun for this edition.

### 83 Limit25

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit
expr = x^2 * (2 + sin(x)^2) / (x + 100)

at x = inf
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 083_limit25.txt | Status: new challenge; not rerun for this edition.

### 84 Limit32

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit

expr = ( sin(x) - cos(x) ) / ( tan(x) - cot(x) )

at x = #pi / 4
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 084_limit32.txt | Status: new challenge; not rerun for this edition.

### 85 Limit35

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit

expr = 1/cos(x) - tan(x)

at x = #pi / 2
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 085_limit35.txt | Status: new challenge; not rerun for this edition.

### 86 Limit40

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit

expr = ( exp(sin(x)) - 1) / x

at x = 0
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 086_limit40.txt | Status: new challenge; not rerun for this edition.

### 87 Limit47

Research challenge. Cancellation, branches or asymptotic behaviour may defeat naive sampling.

```
limit

expr = (10^x - 1) / tan(x)

at x = 0
```

Verification target. Compare one-sided values, domain restrictions and high-precision asymptotic evidence; do not confuse undetermined with nonexistence.

Executable file: 087_limit47.txt | Status: new challenge; not rerun for this edition.

## Compute special functions

### 88 Compute01

Research challenge. Precision, argument domains, identities and function exploration.

```
compute
a = gamma(5)
b = erf(1)
c = besselj(0,2)
d = lambertw(5)
e = zeta(2)
f = a + b + c
```

Verification target. Reference identities: gamma(5)=24; zeta(2)=$\pi^2/6$; the chained sum must use computed variables.

Executable file: 088_compute01.txt | Status: new challenge; not rerun for this edition.

### 89 Compute02

Research challenge. Precision, argument domains, identities and function exploration.

```
compute
function = besselj(0,x)
range = [0,20]
points = 500
```

Verification target. Reference: besselj(0,0)=1 and its roots and oscillatory decay should be represented.

Executable file: 089_compute02.txt | Status: new challenge; not rerun for this edition.

### 90 Compute03

Research challenge. Precision, argument domains, identities and function exploration.

```
compute
function = log(x)
range = [1,20]
points = 20
```

Verification target. Compare against published special-function identities or independent multiprecision evaluation; inspect domain and precision.

Executable file: 090_compute03.txt | Status: new challenge; not rerun for this edition.

### 91 Compute Custom04

Research challenge. Precision, argument domains, identities and function exploration.

```
compute
a = gamma(0.5)
b = erf(1)
c = erfc(1)
d = a*a
e = b+c
```

Verification target. Identities: gamma(1/2)$^2$=π; erf(1)+erfc(1)=1.

Executable file: 091_compute_custom04.txt | Status: new challenge; not rerun for this edition.

### 92 Compute Custom05

Research challenge. Precision, argument domains, identities and function exploration.

```
compute
a = zeta(2)
b = zeta(4)
```

```
c = gamma(6)
d = lambertw(1)
```
Verification target. Identities: zeta(2)=$\pi^2/6$; zeta(4)=$\pi^4/90$; gamma(6)=120.

Executable file: 092_compute_custom05.txt | Status: new challenge; not rerun for this edition.

### 93 Compute Custom06

Research challenge. Precision, argument domains, identities and function exploration.

```
compute
function = erf(x)
range = [-3,3]
points = 301
```
Verification target. Reference: erf is odd; erf(0)=0; the range endpoints should approach ±1.

Executable file: 093_compute_custom06.txt | Status: new challenge; not rerun for this edition.

## Further difficult quadrature

### 94 Quad54

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = exp(-x^2)*cos(10*x)
variable = x
interval = [-inf,inf]
```
Verification target. Exact reference: $\int \exp(-x^2)\cos(10x)\,dx$ over $(-\infty,\infty) = \sqrt{\pi}\exp(-25)$.

Executable file: 094_quad54.txt | Status: new challenge; not rerun for this edition.

### 95 Quad55

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = 1/(1+x)^(5/4)
variable = x
interval = [0,inf]
```
Verification target. Compare with independent high-precision quadrature and check singularity, oscillation, infinite-tail handling, and error flags.

Executable file: 095_quad55.txt | Status: new challenge; not rerun for this edition.

### 96 Quad56

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = exp(-x)-exp(-1.00000001*x)
variable = x
interval = [0,inf]
```
Verification target. Compare with independent high-precision quadrature and check singularity, oscillation, infinite-tail handling, and error flags.

Executable file: 096_quad56.txt | Status: new challenge; not rerun for this edition.

### 97 Quad57

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = 1/(x^(2/3)*(1-x)^(3/4))
variable = x
interval = [0,1]
```

Verification target. Compare with independent high-precision quadrature and check singularity, oscillation, infinite-tail handling, and error flags.

Executable file: 097_quad57.txt | Status: new challenge; not rerun for this edition.

### 98 Quad58

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = sin(100*x)/sqrt(x)
variable = x
interval = [0,1]
```

Verification target. Compare with independent high-precision quadrature and check singularity, oscillation, infinite-tail handling, and error flags.

Executable file: 098_quad58.txt | Status: new challenge; not rerun for this edition.

### 99 Quad59

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = x/sqrt(1-sqrt(x))
variable = x
interval = [0,1]
```

Verification target. Compare with independent high-precision quadrature and check singularity, oscillation, infinite-tail handling, and error flags.

Executable file: 099_quad59.txt | Status: new challenge; not rerun for this edition.

### 100 Quad60

Research challenge. Cancellation, oscillation or improper integration.

```
quadrature
integrand = 1/(1+x^2)^5
variable = x
interval = [0,inf]
```

Verification target. Compare with independent high-precision quadrature and check singularity, oscillation, infinite-tail handling, and error flags.

Executable file: 100_quad60.txt | Status: new challenge; not rerun for this edition.

## Recommended acceptance procedure

1. Run each input using the same current Windows binary and record program version, compiler, time budget and arithmetic mode.
2. For exact integer cases verify substitutions using arbitrary-precision integers and distinguish a representative from the full family.
3. For numerical cases record an independent analytical or multiprecision reference, a toleranced comparison and the solver’s diagnostic status.
4. Track failures, timeouts, unsupported syntax and indeterminate outputs separately; do not label them PASS.
5. For any CSV-dependent model, resolve file paths or replace with inline tabular data before automated execution.