풀이 자습서 > 풀이 자습서 정보

풀이 자습서 정보

PTC Mathcad의 풀이 시스템과 풀이 구간을 사용하여 해를 구하고 최적화를 수행할 수 있습니다. 미지수가 여러 개인 방정식 시스템의 해를 구할 수 있으며, 방정식을 매개변수화하여 매개변수가 해에 미치는 영향을 분석할 수도 있습니다. 선형 및 비선형 방정식과 미분 방정식을 사용할 수 있으며 함수를 최대화하거나 최소화할 수도 있습니다. 풀이 자습서는 다음과 같은 순차적인 3개의 연습으로 구성되어 있습니다.

- 연습 1: 방정식 시스템 풀이 및 함수 근 구하기
- 연습 2: 함수 최적화
- 연습 3: 상미분 방정식 풀기

연습은 순서대로 완료해야 합니다. 연습 1로 이동합니다.

풀이 자습서 > 연습 1 정보

연습 1 정보

PTC Mathcad 풀이 시스템과 풀이 구간을 사용하면 방정식 시스템을 풀고 함수의 근을 구할 수 있습니다. 풀이 시스템은 PTC Mathcad 기본 제공 함수와 유사합니다. 즉, 특정 인수 집합을 필요로 하며 문제를 풀이하는 명령 하나를 제공합니다. 각 풀이 시스템은 특정 문제를 풀도록 설계되어 있습니다. 이와 달리 풀이 구간은 여러 개의 영역으로 구성된 특수한 구역입니다. 각 영역에서 자연스러운 표기법을 사용하여 문제를 정의할 수 있습니다. 풀이 구간을 사용하면 풀이 시스템보다 훨씬 다양한 문제를 풀 수 있습니다. 풀이 구간을 매개변수화하여 나중에 워크시트에서 호출할 수도 있습니다. 이 연습을 마친 후에는 다음과 같은 작업을 수행할 수 있게 됩니다.

- 선형 또는 비선형 방정식 시스템을 풀이할 수 있습니다.
- 함수의 근을 찾을 수 있습니다.
- 풀이 구간을 매개변수화할 수 있습니다.

작업 1-1로 이동합니다.

풀이 자습서 > 작업 1–1: 선형 방정식 시스템

작업 1–1: 선형 방정식 시스템

아래에 정의된 문제를 읽고 다음 방법을 사용하여 해를 구합니다.

- 행렬 계산
- 풀이 시스템
- 풀이 구간

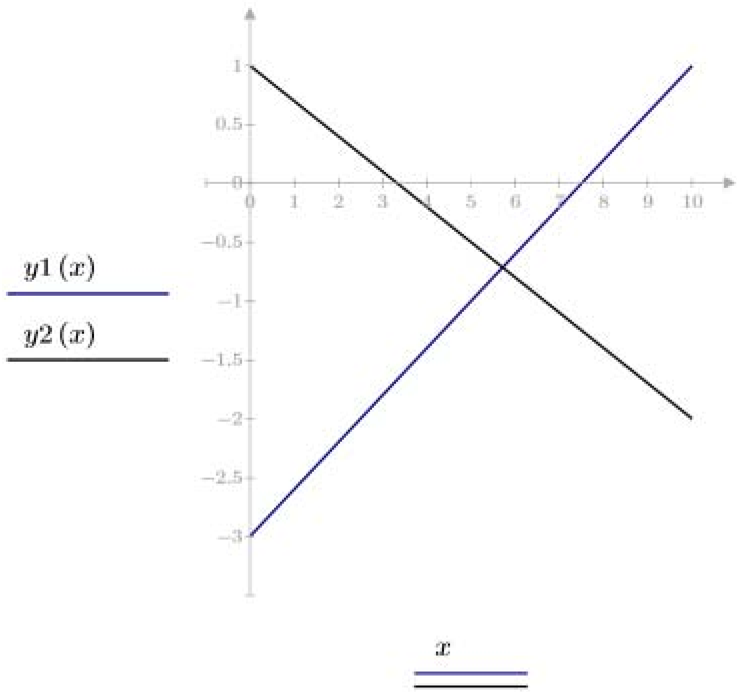
문제 정의

다음 함수들은 선형 함수입니다.

$$y1(x) := \frac{1}{2.5} \cdot (x - 7.5)$$

$$y2(x) := -0.3\ x + 1$$

다음 도표에서 볼 수 있는 것처럼 이 함수들은 교차합니다.



다음 방정식이 참이 되는 교차점의 좌표 *(x, y)*를 구하려고 합니다.

$$y = \frac{1}{2.5} \cdot (x - 7.5)$$

$$y = -0.3\ x + 1$$

방정식을 재배치하여 변수가 방정식의 좌변에 오게 할 수 있습니다.

$$x-2.5\ y=7.5$$

$$0.3\ x+y=1$$

벡터와 행렬을 사용하여 방정식을 다시 작성할 수 있습니다.

$$\begin{bmatrix} 1 & -2.5 \\ 0.3 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7.5 \\ 1 \end{bmatrix}$$

위 방정식의 각 벡터와 행렬을 변수로 표현할 수 있습니다.

$$M \cdot X = v$$

배열 *M* 및 *v*는 알고 있는 값이지만 *X*는 알 수 없는 값입니다. *X*는 요소가 2개인 벡터로, 교차점의 *x* 및 *y* 좌표를 나타냅니다.

행렬 계산으로 풀기

1. 행렬 *M* 및 벡터 *v*를 정의합니다.

$M := \begin{bmatrix} 1 & -2.5 \\ 0.3 & 1 \end{bmatrix}$	$v := \begin{bmatrix} 7.5 \\ 1 \end{bmatrix}$

2. *X*를 행렬 *M*의 역행렬과 벡터 *v*의 곱으로 정의합니다.

$$X := M^{-1} \cdot v$$

3. *X*를 계산합니다.

$$X = \begin{bmatrix} 5.714 \\ -0.714 \end{bmatrix}$$

교차점의 *x* 값은 5.714이고 *y* 값은 -0.714입니다.

풀이 시스템으로 풀기

풀이 시스템은 특정 문제를 푸는 함수입니다. **lsolve** 기본 제공 함수를 사용하여 교차점의 좌표를 구할 수 있습니다.

1. 행렬 *M* 및 벡터 *v*를 정의합니다.

$M := \begin{bmatrix} 1 & -2.5 \\ 0.3 & 1 \end{bmatrix}$	$v := \begin{bmatrix} 7.5 \\ 1 \end{bmatrix}$

2. **lsolve** 함수를 호출합니다.

$$lsolve(M, v) = \begin{bmatrix} 5.714 \\ -0.714 \end{bmatrix}$$

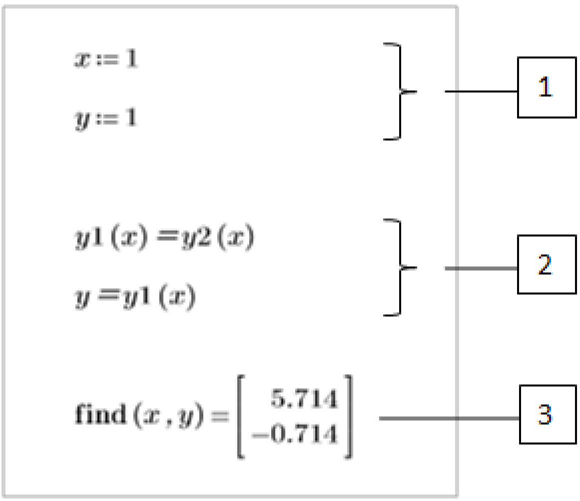
풀이 구간으로 풀기

풀이 구간은 자연스러운 표기법을 사용하여 문제를 정의할 수 있는 영역입니다. 풀이 구간은 행렬 계산이나 풀이 시스템과 달리 방정식을 재배치할 필요가 없습니다. 선형 함수 *y1* 및 *y2*를 다시 호출합니다.

$$y1(x) := \frac{1}{2.5} \cdot (x - 7.5)$$

$$y2(x) := -0.3\, x + 1$$

다음 풀이 구간에서는 **find** 함수를 사용하여 두 함수의 교차점을 계산합니다.

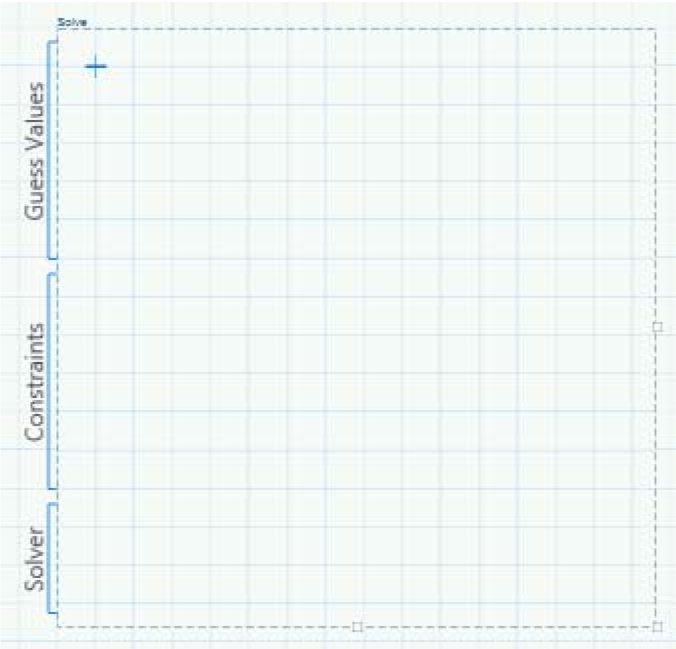
<table>
<tr><td></td><td colspan="3">
</td></tr>
<tr><td></td><td colspan="3">여기서</td></tr>
<tr><td></td><td></td><td>1</td><td>각 미지수에 대한 추측값</td></tr>
<tr><td></td><td></td><td>2</td><td>각 미지수에 대한 제약 조건</td></tr>
<tr><td></td><td></td><td>3</td><td>풀이 구간 함수</td></tr>
</table>

1. 풀이 구간을 사용하여 교차점의 좌표를 찾기 위해 먼저 워크시트에 *y1* 및 *y2*의 두 함수를 정의합니다.

$$y1(x) := \frac{1}{2.5} \cdot (x - 7.5)$$

$$y2(x) := -0.3\, x + 1$$

2. 풀이 구간 영역을 삽입하려면 **수학** 탭의 **영역** 그룹에서 **풀이 구간**을 클릭합니다.



풀이 구간 영역의 크기를 조정하려면 정사각형 핸들 3개 중 하나를 드래그합니다. 워크시트에서 풀이 구간을 이동하면 그 안에 있는 모든 영역이 함께 이동합니다.

3. 풀이 구간 영역에서 문제에 대한 추측값을 입력합니다. PTC Mathcad는 해를 구할 때 이 추측값을 시작점으로 사용합니다.

$$x := 1$$

$$y := 1$$

4. 문제를 제한하는 제약 조건을 입력합니다. 제약 조건을 정의할 때에는 부울 연산자를 사용해야 합니다. "같음" 부울 연산자를 삽입합니다.

$$x := 1$$

$$y := 1$$

$$y1(x) = y2(x)$$

$$y = y1(x)$$

첫 번째 제약 조건은 교차점의 *x* 값을 정의하고, 두 번째 제약 조건은 교차점의 *y* 값을 정의합니다.

5. 풀이 구간 함수 이름과 인수를 삽입합니다. 여기에서는 **find**를 입력한 다음 함수 인수로 *x* 및 *y*를 입력합니다. **find**의 레이블이 자동으로 *keyword*로 설정됩니다.

$$x := 1$$
$$y := 1$$
$$y1(x) = y2(x)$$
$$y = y1(x)$$
$$\mathbf{find}(x, y)$$

6. 풀이 구간을 계산합니다.

$$x := 1$$
$$y := 1$$
$$y1(x) = y2(x)$$
$$y = y1(x)$$
$$\mathbf{find}(x, y) = \begin{bmatrix} 5.714 \\ -0.714 \end{bmatrix}$$

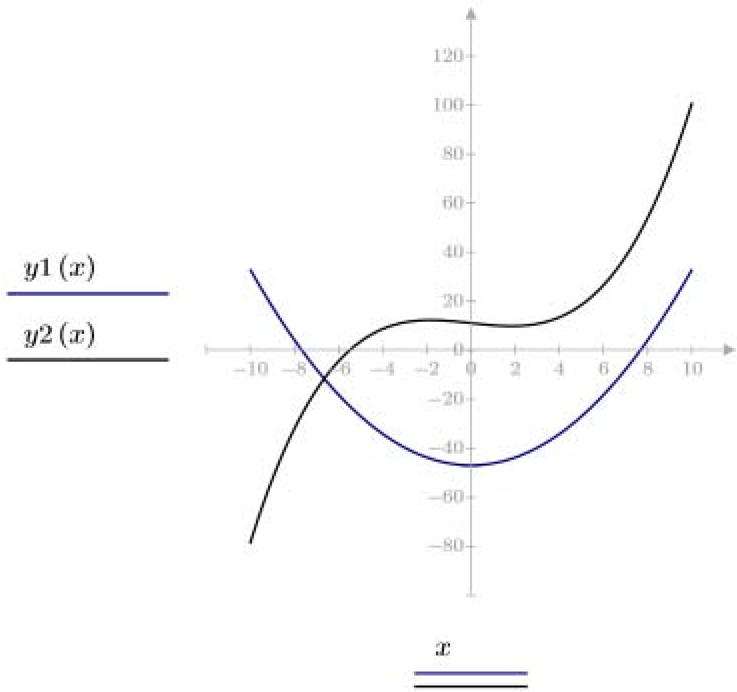
작업 1–2로 이동합니다.

작업 1–2: 비선형 방정식 시스템

이전 작업에서 확인한 것처럼, 풀이 구간은 문제를 쉽게 정의할 수 있는 자연스러운 환경을 제공합니다. 그러므로 풀이 시스템이나 행렬 계산보다 풀이 구간이 훨씬 범용적입니다. 다음 예에서는 두 비선형 함수의 교차점 좌표를 구합니다.

$$y1(x) := -47 + 0.8\ x^2$$

$$y2(x) := 11 - x + 0.1\ x^3$$



풀이 구간으로 풀기

1. Ctrl+1을 눌러 풀이 구간 영역을 삽입한 후 다음 항목을 삽입합니다.
    - *y1* 및 *y2* 함수의 정의
    - 그래프에 기반한 교차점 좌표에 대한 추측값
    - 두 미지수에 대한 두 제약 조건
    - *keyword* 레이블이 자동으로 지정되는 풀이 구간 함수 **find**

$$y1(x) := -47 + 0.8\ x^2$$
$$y2(x) := 11 - x + 0.1\ x^3$$

$$x := -4$$
$$y := 0$$

$$y1(x) = y2(x)$$
$$y = y1(x)$$

$$\text{find}(x, y) = \begin{bmatrix} -6.644 \\ -11.685 \end{bmatrix}$$

풀이 시스템으로 풀기

함수 *y1* 및 *y2*를 다시 호출합니다.

$$y1(x) := -47 + 0.8\ x^2$$

$$y2(x) := 11 - x + 0.1\ x^3$$

새 함수 *f(x) = y2(x) - y1(x)*를 정의할 수 있습니다.

$$f(x) := \left(11 - x + 0.1\ x^3\right) - \left(-47 + 0.8\ x^2\right)$$

$$f(x) := 58 - x - 0.8\ x^2 + 0.1\ x^3$$

새로 정의된 함수 *f*는 비선형 함수의 교차점과 같은 *x* 값에서 x축을 통과합니다. *f*는 다항식이므로 보다 범용적인 **root** 풀이 시스템 대신 **polyroots** 함수를 사용하여 *f*가 x축을 통과하는 위치를 찾습니다.

1. 벡터 *c*에 다항식 계수를 지정합니다. *c*의 첫 번째 요소는 절편이고, 그 다음에는 *x*의 각 거듭곱에 대한 계수가 오름차순으로 나타납니다.

$$c := \begin{bmatrix} 58 \\ -1 \\ -0.8 \\ 0.1 \end{bmatrix}$$

2. **polyroots** 함수를 호출합니다.

$$r := \text{polyroots}(c) = \begin{bmatrix} -6.644 \\ 7.322 + 5.804i \\ 7.322 - 5.804i \end{bmatrix}$$

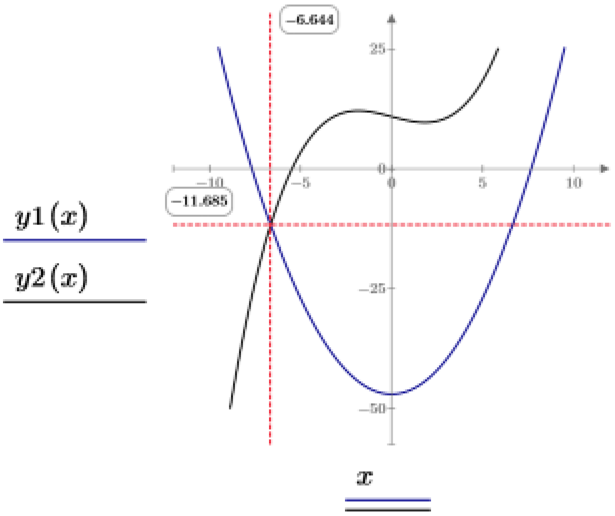
- **polyroots** 함수는 모든 실수 해와 복소수 해로 구성된 벡터를 반환하며, 실수 해가 먼저 나열됩니다.
- 이와 달리 풀이 구간은 한 번에 한 해만 반환합니다. 다른 해를 구하려면 다른 추측값을 사용해야 합니다.

3. 교차점 *(h, v)*의 가로 좌표와 세로 좌표를 계산합니다.

$$h := r_0 = -6.644$$

$$v := \left(y1\left(r_0\right)\right) = -11.685$$

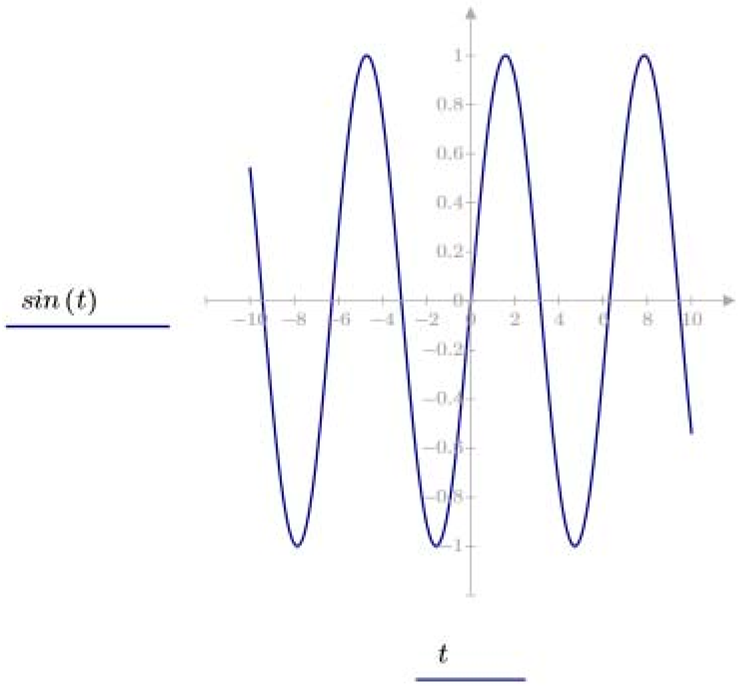
4. 세로 및 가로 마커를 사용하여 도표에 교차점을 표시합니다.



작업 1–3으로 이동합니다.

작업 1–3: 근 구하기

root 풀이 시스템을 사용하여 함수가 x축과 교차하는 위치를 찾습니다. 예를 들어 싸인 곡선 신호의 근 중 일부를 구합니다.



root와 구간 사용

1. **root** 함수를 삽입하려면 **함수** 탭의 **함수** 그룹에서 **풀이**를 클릭합니다. **풀이** 목록이 열립니다. **root**를 선택합니다. **root** 함수의 레이블은 키워드로 표시됩니다.

   $\mathbf{root}(\blacksquare,\blacksquare,\blacksquare,\blacksquare)$

2. 각 자리 표시자에 인수를 입력한 다음 함수를 계산합니다.

   $\mathbf{root}(\sin(x), x, 4, 8) = 6.283$

   root 함수는 지정된 *4 < x < 8* 구간에서 해를 찾습니다. 도표에서 알 수 있는 것처럼 해는 6보다 약간 큽니다.

root를 사용하면 미지수가 하나만 있는 함수의 근을 구할 수 있습니다.

root와 추측값 사용

구간을 사용하는 대신 추측값을 정의하고 **root**를 호출할 수 있습니다. 풀이 구간의 경우 추측값은 **root**가 풀이 루틴을 시작하는 위치입니다.

1. 원점 왼쪽에 있는 근을 구하기 위해 *-4*의 추측값으로 시작합니다.

   $x := -4$

2. 다음 식을 입력합니다.

   $\mathbf{root}(\sin(x), x) = -3.142$

3. 원점 오른쪽에 있는 근을 구하기 위해 새 추측값을 정의합니다.

   $x := 6$

4. 싸인 곡선의 근을 계산합니다. 다른 결과가 반환됩니다.

   $\mathbf{root}(\sin(x), x) = 6.283$

작업 1–4로 이동합니다.

작업 1–4: 풀이 구간 매개변수화

식 복사

풀이 구간은 워크시트의 독립적 영역이 될 수 있지만 워크시트와 상호 작용할 수도 있습니다. 예를 들어, 풀이 구간 영역보다 먼저 나타나는 계산 영역에서 추측값을 정의할 수 있습니다.

$x := 0$

$x^2 = 2$

$\mathbf{find}(x) = 1.414$

풀이 구간 영역 내에서 추측값을 정의하면 해당 영역에만 한정적으로 정의되고, 워크시트의 변수 값에는 영향을 미치지 않습니다.

$x := 0$

$x := 1$

$x^2 = 2$

$\mathbf{find}(x) = 1.414$

$x = 0$

해를 변수에 지정하여 워크시트에서 나중에 사용할 수 있습니다.

$x := 0$

$x^2 = 2$

$r2 := \mathbf{find}(x)$

$r2 = 1.414$

풀이 구간의 매개변수와 동일한 수의 인수를 갖는 함수에 해를 지정할 수 있습니다. 여기서 매개변수는 *a*입니다.

$x := 0$

$x^2 = a$

$f(a) := \mathbf{find}(x)$

함수 *f*를 사용하여 특정 *a* 값에 대한 해를 계산할 수 있습니다.

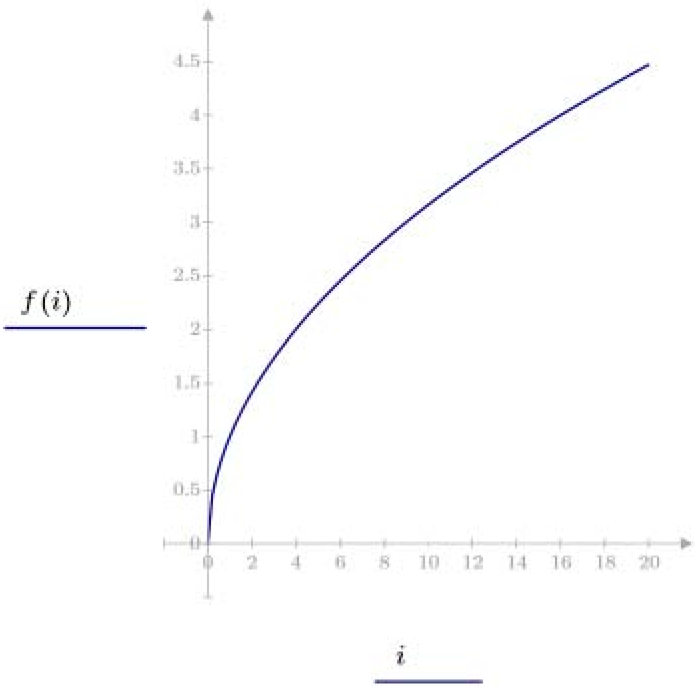
$f(7) = 2.646$

식 복사

$\sqrt{7} = 2.646$

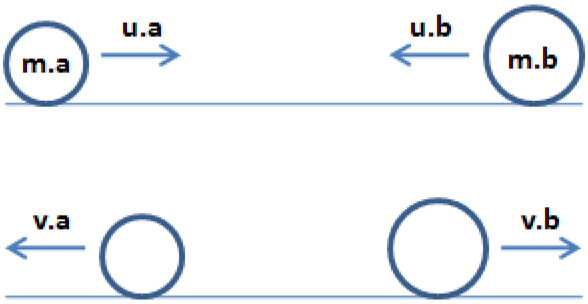
또한 함수 *f*를 도표화하여 매개변수 *a*에 따라 어떻게 변화하는지 시각화할 수 있습니다.

$i := 0, 0.2..20$



실습

다음 연습으로 이동하기 전에 서로 마주 보고 구르다가 충돌하는 두 공의 문제를 살펴보겠습니다.



충돌한 후 공의 속도를 구하는 풀이 구간을 다음과 같이 설정할 수 있습니다.

$m.b$	$u.a$	$u.b$
(kg)	$\left(\frac{m}{s}\right)$	$\left(\frac{m}{s}\right)$
1	1	1

식 복사

$$v.a := 0\ \frac{m}{s} \qquad v.b := 0\ \frac{m}{s}$$

$$m.a \cdot u.b + m.b \cdot (-u.b) = m.a \cdot (-v.a) + m.b \cdot v.b$$

$$\frac{1}{2}\ m.a \cdot u.a^2 + \frac{1}{2}\ m.b \cdot u.b^2 = \frac{1}{2}\ m.a \cdot v.a^2 + \frac{1}{2}\ m.b \cdot v.b^2$$

$$V(m.a) := \mathbf{find}(v.a, v.b)$$

$$\mathrm{v.a}(m.a) := V(m.a)_0$$

$$\mathrm{v.b}(m.a) := V(m.a)_1$$

- 추측값의 단위는 풀이 구간의 해와 호환 가능합니다.
- 제약 조건은 운동량 보존 법칙과 에너지 보존 법칙입니다.
- 풀이 구간 해 *V(m.a)*는 벡터 함수입니다. *v.a* 및 *v.b*는 함수로 레이블이 지정되므로 이후의 계산에서 변수와 구분됩니다.

  충돌 중의 운동량 변화:

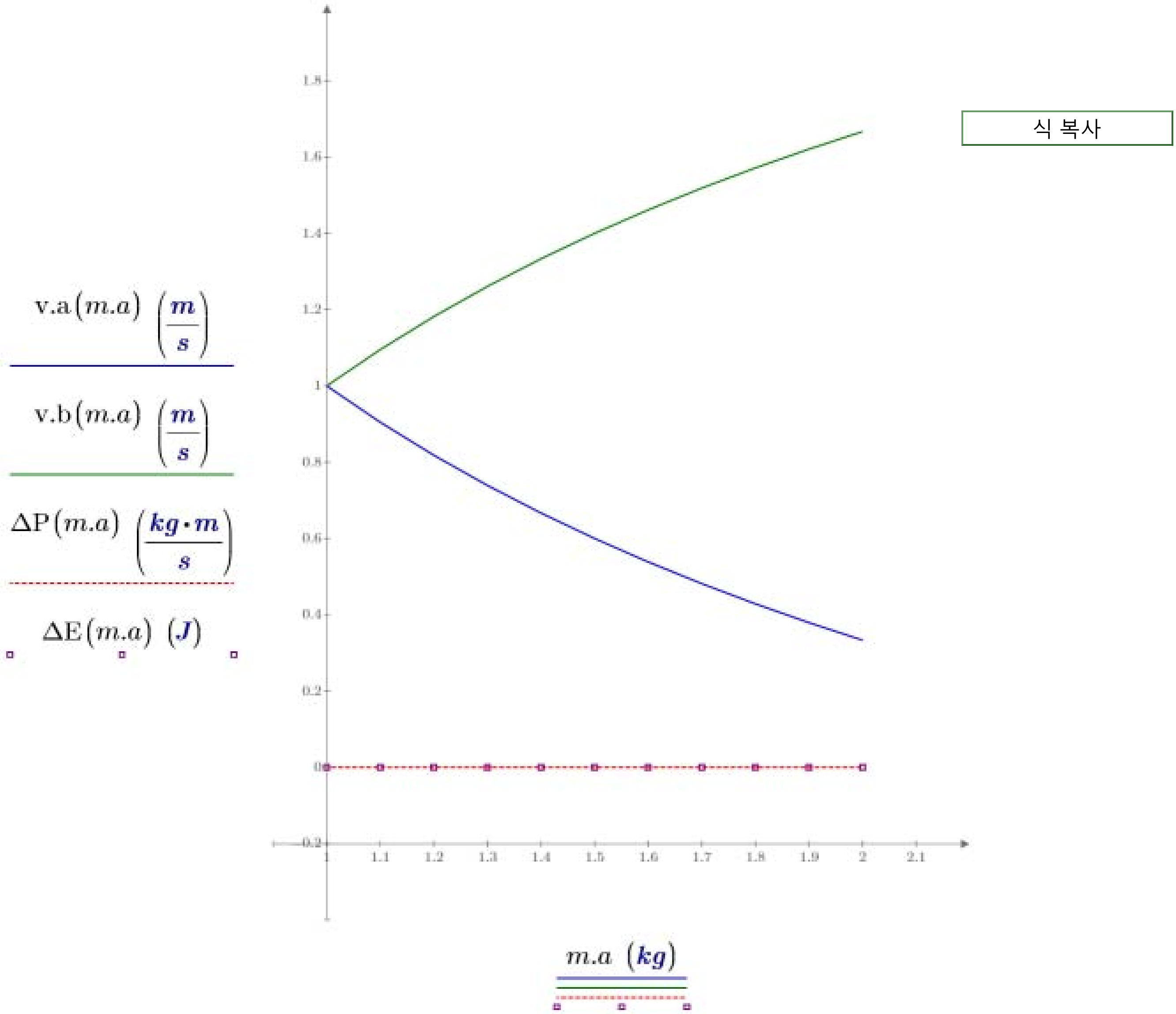
  $$\Delta\mathrm{P}(m.a) := (m.a \cdot u.a + m.b \cdot (-u.b)) - (m.a \cdot (-\mathrm{v.a}(m.a)) + m.b \cdot \mathrm{v.b}(m.a))$$

  충돌 중의 에너지 변화:

  $$\Delta\mathrm{E}(m.a) := \frac{1}{2}\left(m.a \cdot u.a^2 + m.b \cdot u.b^2\right) - \frac{1}{2}\left(m.a \cdot (\mathrm{v.a}(m.a))^2 + m.b \cdot (\mathrm{v.b}(m.a))^2\right)$$

*m.a*를 사용하여 최종 속도와 운동량 및 에너지의 변화를 도표화할 수 있습니다.

$$m.a := 1\ kg, 1.1\ kg .. 2\ kg$$



이 항목의 오른쪽 맨 위에서 **식 복사**를 클릭합니다. 식을 새 워크시트에 붙여 넣으려면 워크시트를 클릭하고 Ctrl+V를 누릅니다. *u.a*의 값을 2로 변경하고 *u.b*의 단위를 ft/s로 변경하여 도표가 어떻게 변하는지 확인합니다.

연습 2로 이동합니다.

풀이 자습서 > 연습 2 정보

연습 2 정보

풀이 구간을 사용하여 최적화를 수행할 수 있습니다. 함수를 최소화하거나 최대화하는 매개변수를 구하고 최적화를 제한하는 제약 조건을 추가할 수 있습니다. 이 연습을 마친 후에는 다음과 같은 작업을 수행할 수 있게 됩니다.

- 함수를 최대화하거나 최소화할 수 있습니다.
- 함수를 최적화할 때 제약 조건을 추가할 수 있습니다.

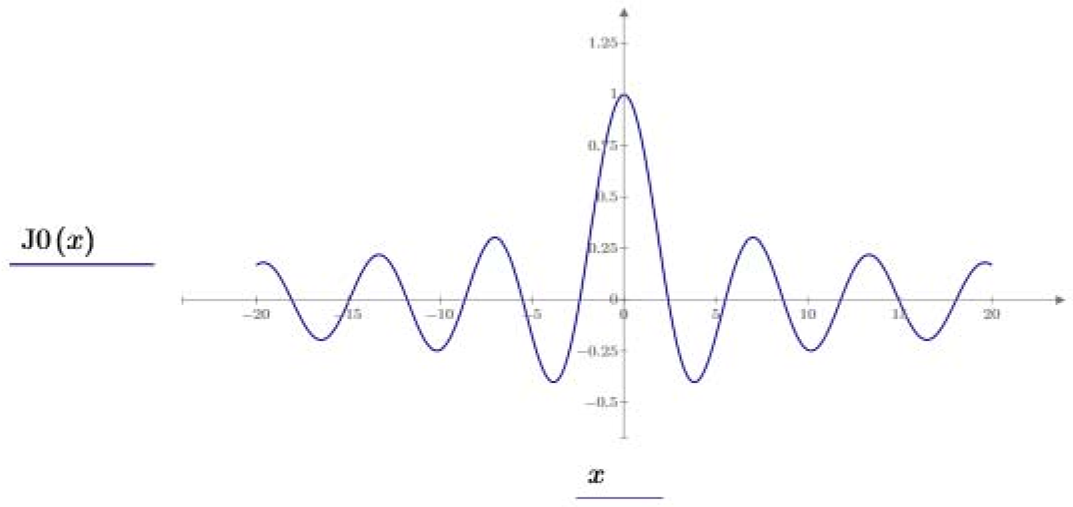
작업 2-1로 이동합니다.

작업 2–1: 함수 최적화

풀이 구간을 사용하여 제1종 0차 베셀 함수 **J0**의 최대 점 수를 구합니다.

가능한 경우 최적화할 함수를 도표로 표시하는 것이 좋습니다. 그러면 적절한 추측값을 더 쉽게 선택할 수 있습니다.

1. **J0** 함수를 도표화합니다.



   J0 함수에는 많은 최대 점과 최소 점이 있습니다. 추측값을 지정하면 가장 가까운 점을 구할 수 있습니다.

2. 풀이 구간을 삽입하고, 최대값의 추측값을 *x1=5*로 정의한 다음, **maximize** 함수를 사용하여 *x1* 주위에서 최대값을 구합니다.

$$x1 := 5$$

$$max1 := \mathbf{maximize}(\mathrm{J0}, \mathbf{x1})$$

   find 함수와 달리 **J0** 함수는 인수 목록 없이 입력해야 합니다.

3. 풀이 구간 외부에서 x_{max1} 및 $J0(x_{max1})$을 계산하여 첫 번째 최대값의 가로 및 세로 좌표를 계산합니다.

$$h1 := max1 = 7.016$$

$$v1 := \mathrm{J0}(max1) = 0.3$$

4. 추측값을 변경하고 해당하는 최대값을 구합니다.

$x2 := -15$

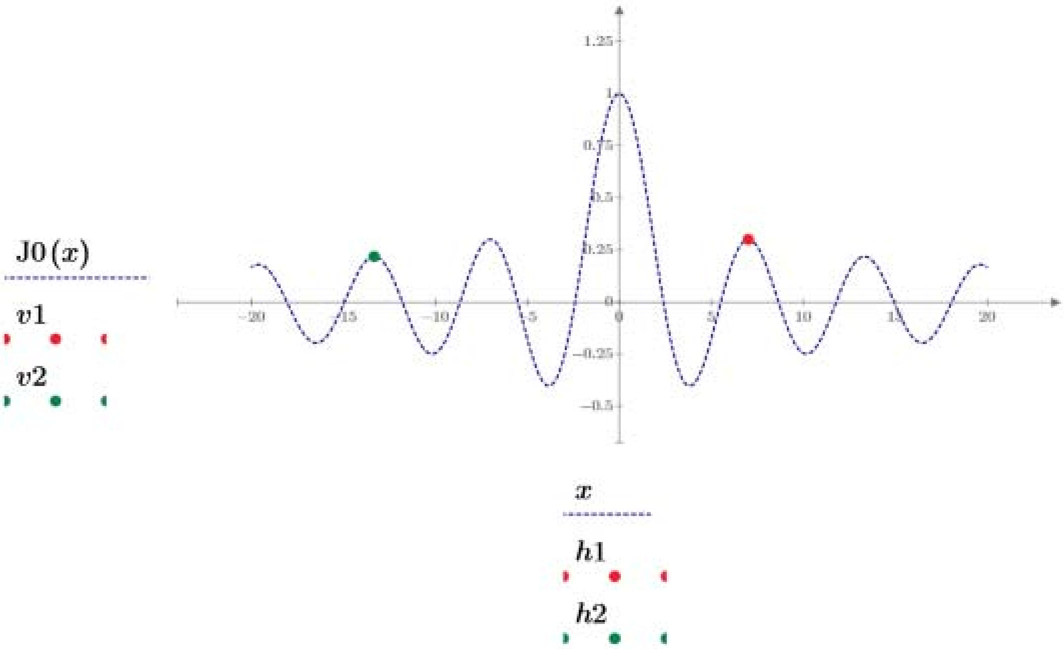
$max2 := \mathbf{maximize}(\mathbf{J0}, \mathbf{x2})$

5. 풀이 구간 외부에서 x_{max2} 및 $J0(x_{max2})$을 계산하여 두 번째 최대값의 가로 및 세로 좌표를 계산합니다.

$h1 := max2 = -13.324$

$v1 := \mathrm{J0}(max2) = 0.218$

6. 원래 도표에 두 최대 점을 표시합니다.



풀이 구간 외부에서 **maximize** 함수 사용

제약 조건을 지정할 필요가 없는 경우 풀이 구간 외부에서 **maximize** 함수를 사용할 수 있습니다.

1. 첫 번째 추측값을 입력하고 해당하는 최대 점을 다시 계산합니다.

$x1 := 5$

$max1 := \mathbf{maximize}(\mathbf{J0}, x1) = 7.016$

2. 두 번째 추측값을 입력하고 해당하는 최대 점을 다시 계산합니다.

$x2 := -15$

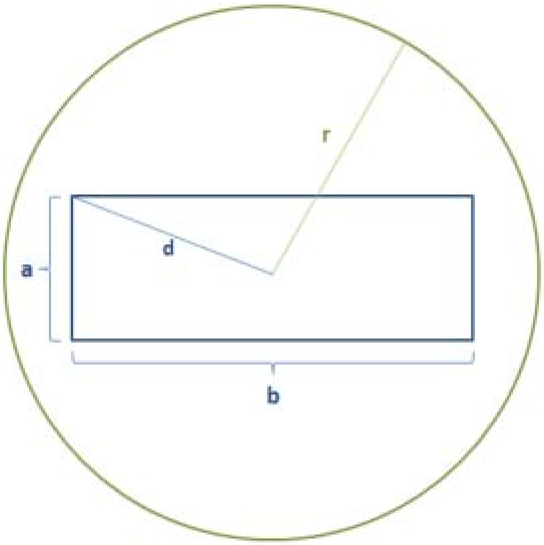
$max2 := \mathbf{maximize}(\mathbf{J0}, x2) = -13.324$

maximize 함수에서 동일한 최대 점이 반환됩니다.

작업 2–2로 이동합니다.

작업 2–2: 제약 조건이 있는 최적화

풀이 구간을 사용하여 원으로 둘러싸인 사각형의 면적을 최대로 만드는 너비와 길이를 구합니다.



1. 원의 반지름을 정의합니다.

   $r := 2$

2. 위 그림에서 볼 수 있는 것처럼 길이 *d*를 정의합니다.

   $$d(a, b) := \frac{\sqrt{a^2 + b^2}}{2}$$

3. 풀이 구간을 삽입하고, *a* 및 *b*의 추측값을 정의하고, 면적 함수를 정의한 다음 직사각형을 원 안쪽에 유지하는 *d* < *r* 제약 조건을 정의합니다. *a* 및 *b*의 해를 구하기 위해 **maximize** 함수를 호출합니다.

   $a := 5$

   $b := 5$

   $area(a, b) := a \cdot b$

   $d(a, b) < r$

   $$\begin{bmatrix} A \\ B \end{bmatrix} := \mathbf{maximize}(area, a, b)$$

4. *A*, *B* 및 *d*를 계산합니다.

   $A = 2.828$

$B=2.828$

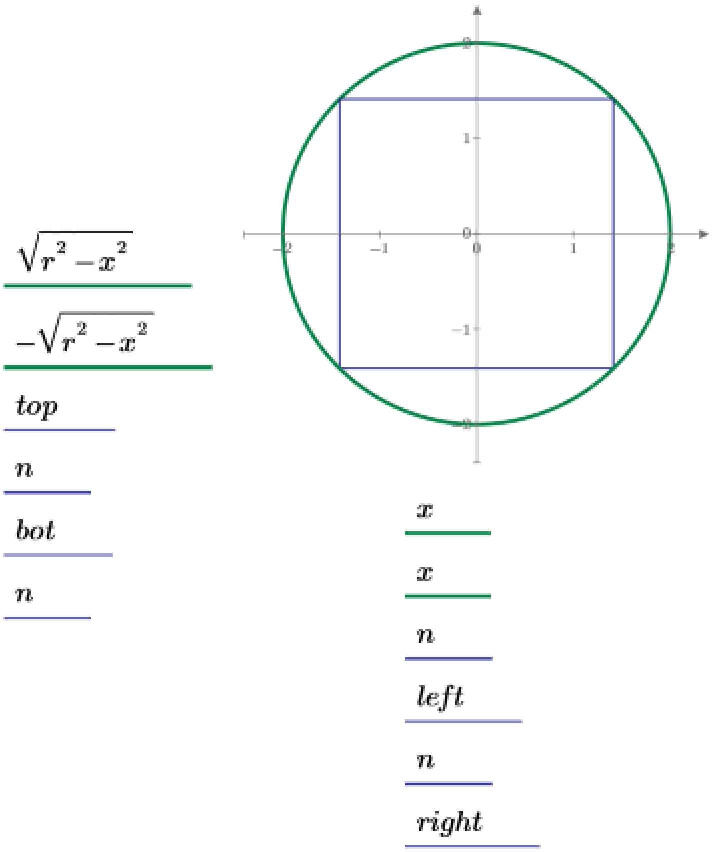
$d(A,B)=2$

예상대로 *A* = *B*입니다. 이것은 최대 면적을 갖는 직사각형은 사실 *d* = *r*인 정사각형이라는 의미입니다.

5. 변 *A* 및 *B*를 사용하여 정사각형 주위에 원을 그립니다.

- 별도의 그래프선을 사용하여 원의 위쪽 절반과 아래쪽 절반을 그립니다.
- 마찬가지로 별도의 그래프선 네 개를 사용하여 정사각형의 네 변을 그립니다.

$n:=\frac{-A}{2},\frac{-A}{2}+0.014..\frac{A}{2}$ $top:=1.414$ $bot:=-1.414$ $left:=-1.414$ $right:=1.414$



작업 2-3으로 이동합니다.

작업 2–3: 비선형 최소자승적합법

식 복사

데이터 집합을 모델링하는 함수의 매개변수를 사용하여 적합식을 정의한 후, 풀이 구간을 사용하여 데이터 집합과 적합식 간의 잉여(residual)를 최소화합니다. 다른 최적화 문제와 마찬가지로 문제를 재배치하여 근을 찾을 수 있습니다. 여기에서는 잉여를 0으로 설정합니다.

1. 데이터 집합을 정의합니다.

$$u:=\begin{bmatrix}0.132\\0.322\\0.511\\0.701\\0.891\\1.081\\1.27\\1.46\\1.65\\1.839\\2.029\\2.219\end{bmatrix}\quad v:=\begin{bmatrix}0.1\\0.258\\0.543\\0.506\\0.606\\0.622\\0.569\\0.453\\0.438\\0.316\\0.29\\0.195\end{bmatrix}$$

2. 미지의 매개변수 α 및 β를 사용하여 *Weibull* 적합식을 정의합니다.

$$Wb(u,\alpha,\beta):=\alpha\cdot\beta\cdot u^{\beta-1}\cdot\exp\left(-\alpha\cdot u^{\beta}\right)$$

3. 데이터 집합의 *v* 값과 *Wb*로 계산된 *v* 값의 차이인 잉여를 정의합니다.

$$resid(\alpha,\beta):=v-\overrightarrow{Wb(u,\alpha,\beta)}$$

4. 제곱의 합을 정의합니다.

$$SSE(\alpha,\beta):=\sum resid(\alpha,\beta)^{2}$$

5. *Weibull* 함수를 가장 적합하게 맞추는 매개변수 α 및 β를 구하기 위해 풀이 구간을 삽입하고 α 및 β의 추측값을 정의한 다음 **minimize** 함수를 호출합니다.

$$\alpha:=0.8\qquad\beta:=1$$

$$\begin{bmatrix}\alpha1\\\beta1\end{bmatrix}:=\text{minimize}(SSE,\alpha,\beta)$$

6. 해를 계산합니다.

$$\begin{bmatrix}\alpha1\\\beta1\end{bmatrix}=\begin{bmatrix}0.502\\2\end{bmatrix}$$

7. 평균 제곱 오차를 계산합니다. 참인 해가 존재하면 이 값은 0이 됩니다.

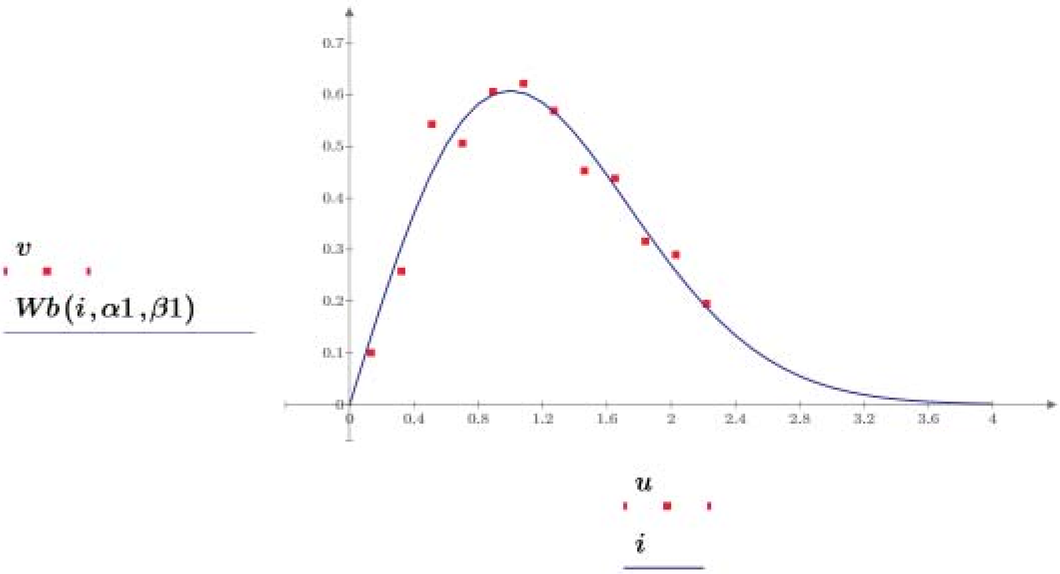
$n := \text{length}(u) - 1$

$\dfrac{SSE(\alpha 1, \beta 1)}{n-2} = 0.002$

식 복사

8. 데이터 집합과 *Weibull* 적합식을 도표화합니다.

$i := 0, 0.1 .. 4$



9. 제약 조건 *resid = 0*을 사용하여 적합식의 매개변수를 구하기 위해 **minimize** 함수 대신 **minerr** 함수를 사용합니다.

$\alpha := 0.8 \qquad \beta := 1$

$resid(\alpha, \beta) = 0$

$\begin{bmatrix} \alpha 2 \\ \beta 2 \end{bmatrix} := \text{minerr}(\alpha, \beta)$

α2 및 β2에 대한 정확한 해가 없기 때문에 여기서는 **find** 함수를 사용할 수 없습니다. 사용하는 경우 해가 없다는 오류가 반환됩니다. **minerr** 함수는 **find** 함수와 동일한 방식으로 작동하지만, 설정된 반복 횟수 내에서 해로 수렴하지 못한 경우 근사해를 구한다는 차이점이 있습니다.

10. 새 매개변수에 대한 평균 제곱 오차를 계산합니다.

$\dfrac{SSE(\alpha 2, \beta 2)}{n-2} = 0.002$

11. **minimize** 및 **minerr**로 구한 결과를 비교합니다.

$\begin{bmatrix} \alpha 2 - \alpha 1 \\ \beta 2 - \beta 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.0000000014 \end{bmatrix}$

실습

다음 연습으로 이동하기 전에 수익을 최대화할 수 있는 물품의 가격을 구합니다($n \cdot p$). n 함수를 사용하여 판매된 물품의 수와 가격 간의 관계를 정의합니다.

식 복사

$$n(p) := 100\ (p - 10)^2 + 1000$$

추측값을 선택하기 전에 $0 < p < 10$에 대해 수익 함수를 도표화합니다.

연습 3으로 이동합니다.

연습 3 정보

풀이 시스템이나 풀이 구간을 사용하여 Stiff 및 비 Stiff 상미분 방정식(ODE)을 풀 수 있습니다. 또한, 야코비를 계산할 수 있습니다. 이 연습을 마친 후에는 다음과 같은 작업을 수행할 수 있게 됩니다.

- Stiff 및 비 Stiff ODE를 풀 수 있습니다.
- 상태-공간에서 ODE를 모델링할 수 있습니다.
- ODE를 매개변수화할 수 있습니다.
- 야코비를 사용할 수 있습니다.

작업 3-1로 이동합니다.

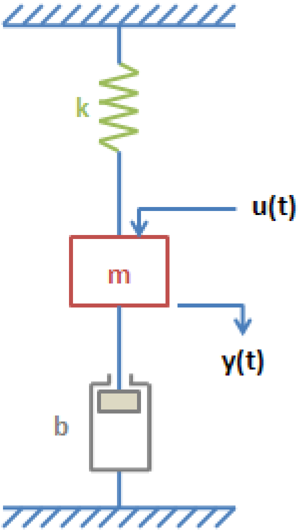
작업 3–1: 상태-공간에서 ODE 모델링

아래에 정의된 문제를 읽고 작업 3-1부터 작업 3-3까지 다음 방법을 사용하여 해를 구합니다.

- 상태-공간 ODE 풀이 시스템
- ODE 풀이 시스템
- 풀이 구간

문제 정의

전형적인 질량-스프링-댐퍼 시스템이 있다고 가정합니다.



이 시스템의 동역학 방정식은 다음과 같습니다.

$$m \cdot y''(t) + b \cdot y'(t) + k \cdot y(t) = u(t)$$

이 시스템을 상태-공간 모델로 표현하면 다음과 같은 형식이 됩니다.

$$\frac{d}{dt} x(t) = A(t) \cdot x(t) + B(t) \cdot u(t)$$

$$y(t) = C(t) \cdot x(t) + D(t) \cdot u(t)$$

여기서

- A - 상태 행렬
- B - 입력 행렬
- C - 출력 행렬
- D - 직접 전달 행렬
- x - 상태 벡터
- u - 입력

- y - 측정 또는 제어되는 출력

시스템 동역학을 모델링하는 상태 및 출력 비선형 방정식을 선형화하여 위의 선형 시스템을 얻을 수 있습니다.

이 2차 방정식 시스템에 상태 변수 두 개를 사용합니다.

$$x_1(t) = y(t)$$

$$x_2(t) = y'(t)$$

m = 1, *b = 0.5* 및 *k = 3*인 경우 시스템 방정식은 다음과 같습니다.

$$x'_1(t) = x_2(t)$$

$$x'_2(t) = -3 \cdot x_1(t) - 0.5 \cdot x_2(t) + u(t)$$

상태-공간 행렬 형식에서 모델은 다음과 같이 작성됩니다.

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3 & -0.5 \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

상태-공간 ODE 풀이 시스템

1. 행렬 함수 *A*, *B*, *C* 및 *D*를 정의합니다.

$A(t) := \begin{bmatrix} 0 & 1 \\ -3 & -0.5 \end{bmatrix}$	$B(t) := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
$C(t) := \begin{bmatrix} 1 & 0 \end{bmatrix}$	$D(t) := 0$

2. 입력이 헤비사이드 계단 함수가 되도록 정의합니다. 계단 함수를 삽입하려면 F 키를 누른 다음 Ctrl+G를 누릅니다.

   $$u(t) := \Phi(t)$$

3. 두 변수의 초기 조건을 정의합니다. 문자식 아래 첨자로 *i*를 입력하려면 **수학** 탭의 **스타일** 그룹에서 **아래 첨자**를 클릭한 다음 i를 입력합니다.

   $$x_i := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

4. 시스템 해를 구할 시간 경계를 정의합니다.

$t_i := 0$

$t_f := 30$

5. t_i를 제외하고 해를 구할 점의 수를 정의합니다.

$npoints := 500$

6. **statespace** 함수를 호출합니다.

$sol := statespace\left(x_i, t_i, t_f, npoints, A, B, u\right)$

행렬 *sol*의 첫 번째 열에는 해를 구한 시간이 포함되고, 나머지 열에는 해당 시간에서의 상태 변수 *x1* 및 *x2*가 포함됩니다.

7. 행렬 *sol*에서 *t*, *x1* 및 *x2*를 추출합니다.

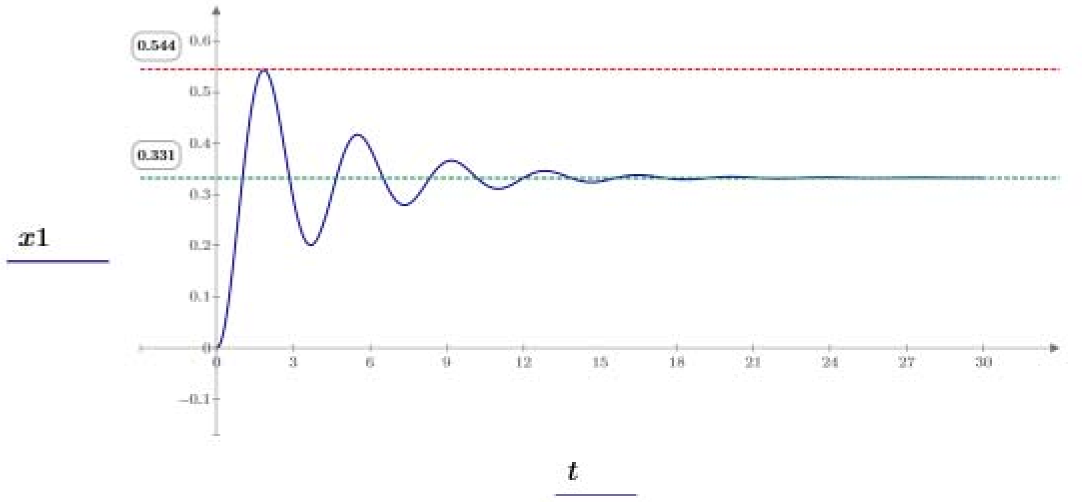
$t := sol^{\langle 0 \rangle}$	$x1 := sol^{\langle 1 \rangle}$	$x2 := sol^{\langle 2 \rangle}$
$t = \begin{bmatrix} 0 \\ 0.06 \\ 0.12 \\ 0.18 \\ 0.24 \\ 0.3 \\ 0.36 \\ 0.42 \\ 0.48 \\ 0.54 \\ 0.6 \\ 0.66 \\ \vdots \end{bmatrix}$	$x1 = \begin{bmatrix} 0 \\ 0.002 \\ 0.007 \\ 0.016 \\ 0.027 \\ 0.042 \\ 0.059 \\ 0.079 \\ 0.101 \\ 0.124 \\ 0.149 \\ 0.176 \\ \vdots \end{bmatrix}$	$x2 = \begin{bmatrix} 0 \\ 0.059 \\ 0.116 \\ 0.169 \\ 0.22 \\ 0.266 \\ 0.309 \\ 0.346 \\ 0.379 \\ 0.407 \\ 0.43 \\ 0.448 \\ \vdots \end{bmatrix}$

8. *x1*의 평균과 최대값을 계산합니다.

$mn := \text{mean}\left(x1\right) = 0.331$

$mx := \max\left(x1\right) = 0.544$

9. 시간에 대해 *x1*을 도표화하고 마커를 사용하여 평균과 최대값을 표시합니다.



도표는 상승 시간, 오버슈트 및 정착 시간과 같은 과도 응답 특성을 보여줍니다.
작업 3–2로 이동합니다.

작업 3–2: ODE 풀이 시스템으로 ODE 풀기

이전 작업에서는 질량-스프링-댐퍼 시스템을 상태-공간 ODE 풀이 시스템을 사용하여 풀었습니다. 여기서는 이 문제를 ODE 풀이 시스템을 사용하여 풉니다. 동역학 방정식은 다음과 같았습니다.

$$m \cdot y''(t) + b \cdot y'(t) + k \cdot y(t) = u(t)$$

시스템 매개변수는 $m = 1$, $b = 0.5$ 및 $k = 3$이고, 입력은 헤비사이드 계단 함수 $u(t) = \Phi(t)$이었습니다. 이 2차 방정식을 1차 ODE로 다시 작성할 수 있습니다.

$$x1'(t) = x2(t)$$

$$x2'(t) = -3 \cdot x1(t) - 0.5 \cdot x2(t) + \Phi(t)$$

1. 시스템의 우변을 지정하는 벡터 함수를 정의합니다.

$$D(t, X) := \begin{bmatrix} X_1 \\ -3 \cdot X_0 - 0.5 \cdot X_1 + \Phi(t) \end{bmatrix}$$

   D의 인수는 독립 변수인 t와 종속 변수의 벡터인 X입니다.

$$X(t) = \begin{bmatrix} x1(t) \\ x2(t) \end{bmatrix}$$

2. $x1$ 및 $x2$에 대한 초기값을 정의합니다.

$$init := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

3. 해를 계산할 최초 시간과 최종 시간을 정의합니다.

$$Ti := 0$$

$$Tf := 30$$

4. 시간 단계의 수를 정의합니다.

$$N := 500$$

5. **AdamsBDF** 풀이 시스템을 호출하여 해를 계산합니다.

$$Sol := AdamsBDF(init, Ti, Tf, N, D)$$

$$Sol = \begin{bmatrix} 0 & 0 & 0 \\ 0.06 & 0.002 & 0.059 \\ 0.12 & 0.007 & 0.116 \\ 0.18 & 0.016 & 0.169 \\ 0.24 & 0.027 & 0.22 \\ 0.3 & 0.042 & 0.266 \\ 0.36 & 0.059 & 0.309 \\ & & \vdots \end{bmatrix}$$

- **AdamsBDF** 풀이 시스템은 복합형 풀이 시스템입니다. 즉, 처음에는 비 Stiff **Adams** 풀이 시스템으로 시작되지만 문제가 Stiff 형식인 경우 자동으로 Stiff **BDF** 풀이 시스템으로 전환됩니다.
- **AdamsBDF** 풀이 시스템을 다른 ODE 풀이 시스템으로 대체할 수도 있습니다. 자세한 내용은 도움말에서 "미분 방정식 풀이 정보" 항목을 참조하십시오.
- 해는 각 N 단계에 대한 시스템의 시간, 변위 및 속도를 나타내는 3열 행렬입니다.

6. Sol에서 시간과 변위를 추출하여 도표화합니다.

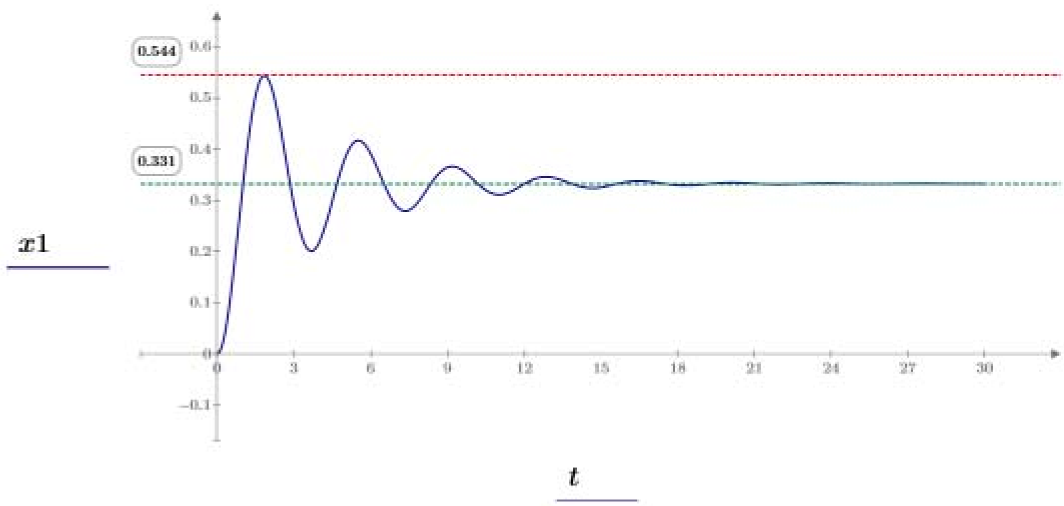
$$t := Sol^{\langle 0 \rangle}$$

$$x := Sol^{\langle 1 \rangle}$$

7. x의 평균과 최대값을 계산합니다.

$$mn := \mathrm{mean}(x) = 0.331 \qquad mx := \max(x) = 0.544$$

8. 시간에 대해 x를 도표화하고 마커를 사용하여 평균과 최대값을 표시합니다.



도표는 상승 시간, 오버슈트 및 정착 시간과 같은 과도 응답 특성을 보여줍니다.

작업 3–3으로 이동합니다.

풀이 자습서 > 작업 3–3: 풀이 구간으로 ODE 풀기

작업 3–3: 풀이 구간으로 ODE 풀기

풀이 구간에서 방정식 시스템을 푸는 것처럼 자연스러운 표기법을 사용하여 ODE를 풀 수 있습니다. 풀이 구간과 새 입력 함수를 사용하여 질량-스프링-댐퍼 시스템을 풉니다.

1. 질량 *m*, 댐핑 계수 *c* 및 스프링 상수 *k*를 정의합니다.

$$m := 2$$

$$c := 2$$

$$k := 8$$

2. 입력 함수 *u(t)*를 정의합니다.

$$u(t) := \frac{1}{2} \cdot cos\left(\frac{3 \cdot \pi}{2} \cdot t\right)$$

3. 다음 풀이 구간을 입력합니다. **수학** 탭의 **연산자 및 기호** 그룹에서 **연산자**를 클릭한 다음 **프라임 연산자**를 클릭하고 *x*의 도함수를 입력합니다. 문제의 초기 조건을 정의한 다음 **odesolve** 함수를 호출합니다.

$$m \cdot x''(t) + c \cdot x'(t) + k \cdot x(t) = u(t)$$

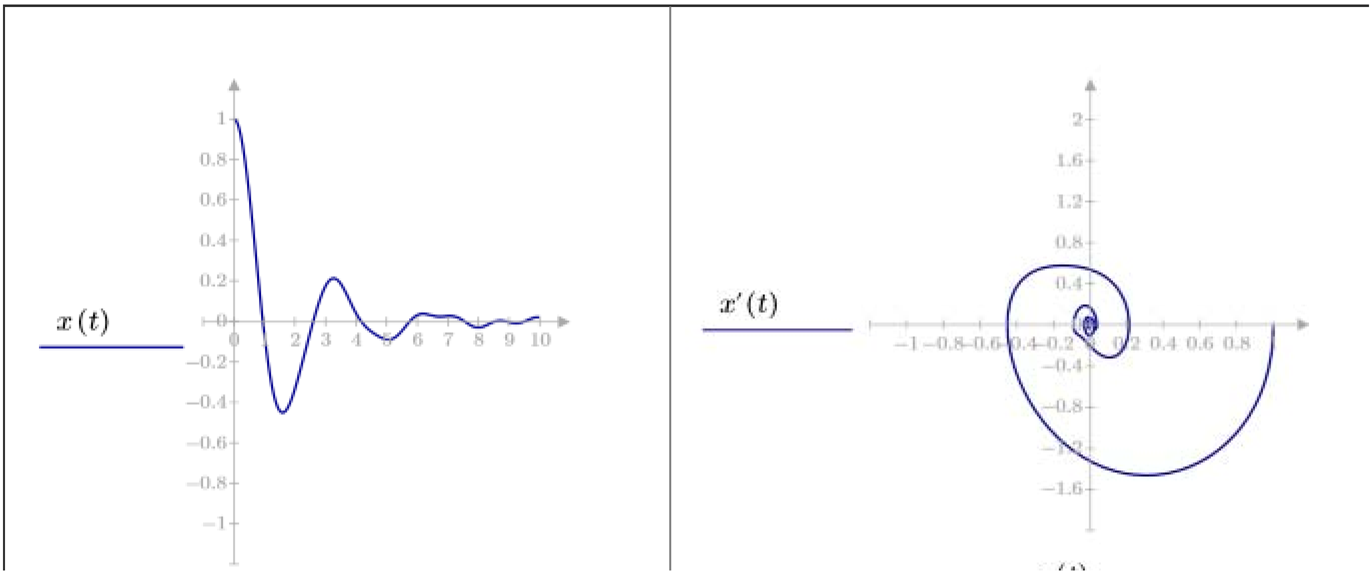
$$x(0) = 1$$

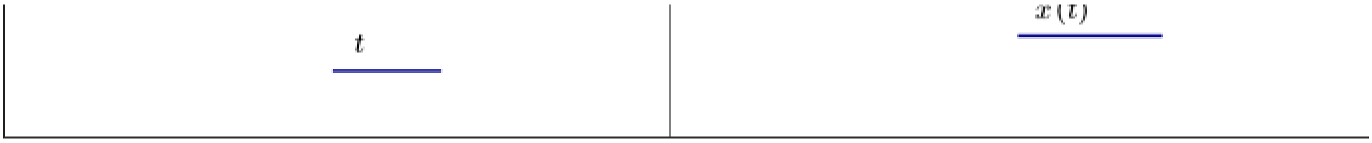
$$x'(0) = 0$$

$$x := \mathrm{odesolve}(x(t), 20)$$

풀이 구간에서 ODE를 풀 경우 추측값을 사용하는 대신 문제의 초기 조건과 경계 조건을 정의해야 합니다.

4. *0 < t < 10* 범위에 대해 해를 도표화합니다.





ODE 매개변수화

1. 풀이 구간을 복사하여 워크시트의 새 위치에 붙여 넣습니다.
2. 초기 조건을 매개변수화합니다. 함수 정의에 매개변수당 하나의 인수를 추가해야 합니다. 여기에서는 *y(a, b)*를 정의합니다.

$$m \cdot x''(t) + c \cdot x'(t) + k \cdot x(t) = u(t)$$

$$x(0) = a$$

$$x'(0) = b$$

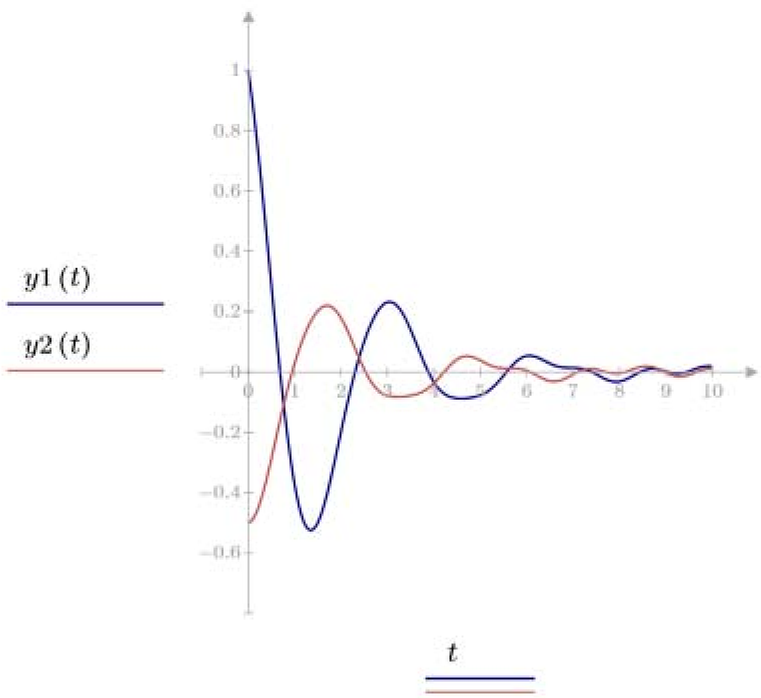
$$y(a, b) := \mathrm{odesolve}(x(t), 20)$$

3. 초기 조건을 달리하여 두 함수를 정의합니다.

$$y1 := y(1, -1)$$

$$y2 := y(-0.5, 0)$$

4. 두 함수를 도표화합니다.



작업 3-4로 이동합니다.

작업 3–4: 풀이 구간을 사용하여 다중 ODE 풀기

Van der Pol 방정식의 해를 구합니다. 이 방정식은 비선형 스프링 시스템의 위치와 속도를 설명합니다.

1. 시스템 매개변수 ε와 풀이를 종료할 시간을 정의합니다.

$\varepsilon := 0.5$

$T := 12$

2. 풀이 구간을 삽입하고 *Van der Pol* 방정식을 입력합니다. **odesolve** 함수를 사용하여 시간의 함수로 X 및 Y의 해를 구합니다.

$$x'(t) = \varepsilon \cdot \left(1 - y(t)^2\right) \cdot x(t) - y(t)$$

$$y'(t) = x(t)$$

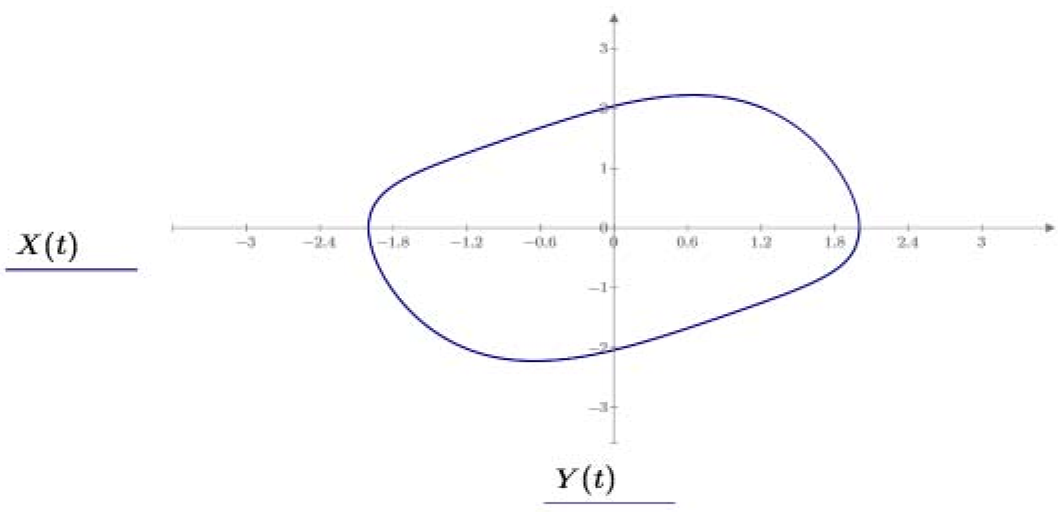
$$x(0) = 0$$

$$y(0) = 2$$

$$\begin{bmatrix} X \\ Y \end{bmatrix} := \text{odesolve}\left(\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, T\right)$$

3. 해를 도표화합니다.

$t := 0, 0.01 .. 12$



4. 풀이 구간을 복사하여 붙여 넣고 초기 조건을 매개변수화합니다.

$$x'(t) = \varepsilon \cdot \left(1 - y(t)^2\right) \cdot x(t) - y(t)$$

$$y'(t) = x(t)$$

$$x(0) = a$$

$$y(0) = b$$

$$F(a, b) := \text{odesolve}\left(\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, T\right)$$

5. 여러 초기 조건에 대한 해를 추출합니다.

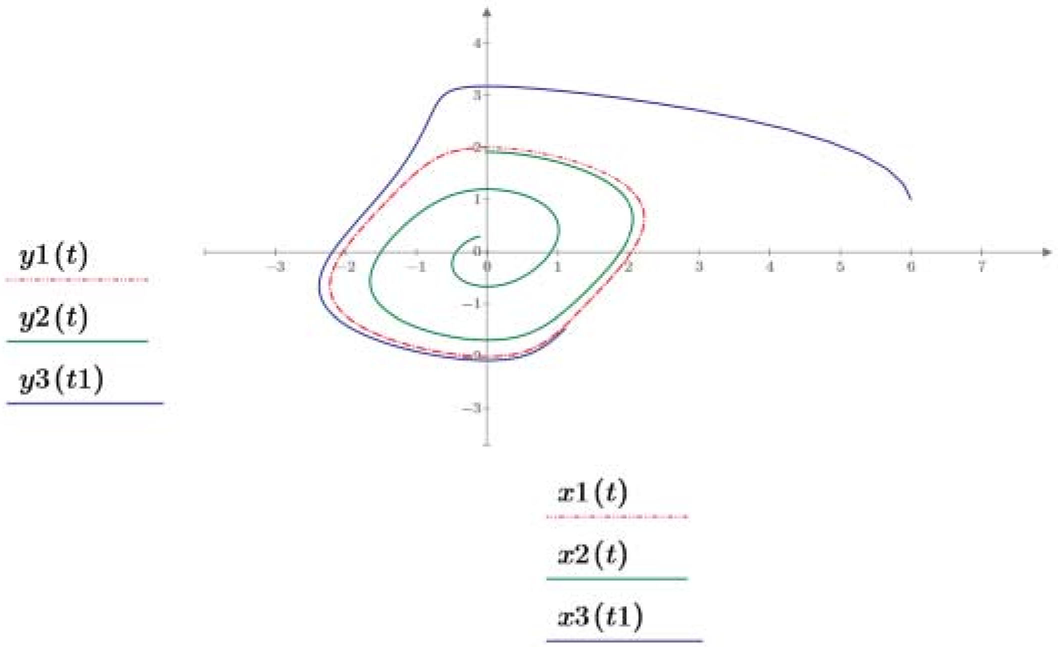
$$\begin{bmatrix} x1 \\ y1 \end{bmatrix} := F(0, 2)$$

$$\begin{bmatrix} x2 \\ y2 \end{bmatrix} := F(-0.1, 0.3)$$

$$\begin{bmatrix} x3 \\ y3 \end{bmatrix} := F(6, 1)$$

6. 해를 도표화합니다.

$$t1 := 0, 0.05 .. 6$$



다른 모든 해가 나선형으로 주기적 해(빨간색)에 접근합니다.

7. 초기 풀이 구간을 복사하여 붙여 넣고 시스템 매개변수 ε을 매개변수화합니다.

$$x'(t) = \varepsilon \cdot \left(1 - y(t)^2\right) \cdot x(t) - y(t)$$

$$y'(t) = x(t)$$

$$x(0) = 0$$

$$y(0) = 2$$

$$F(\varepsilon) := \text{odesolve}\left(\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, T\right)$$

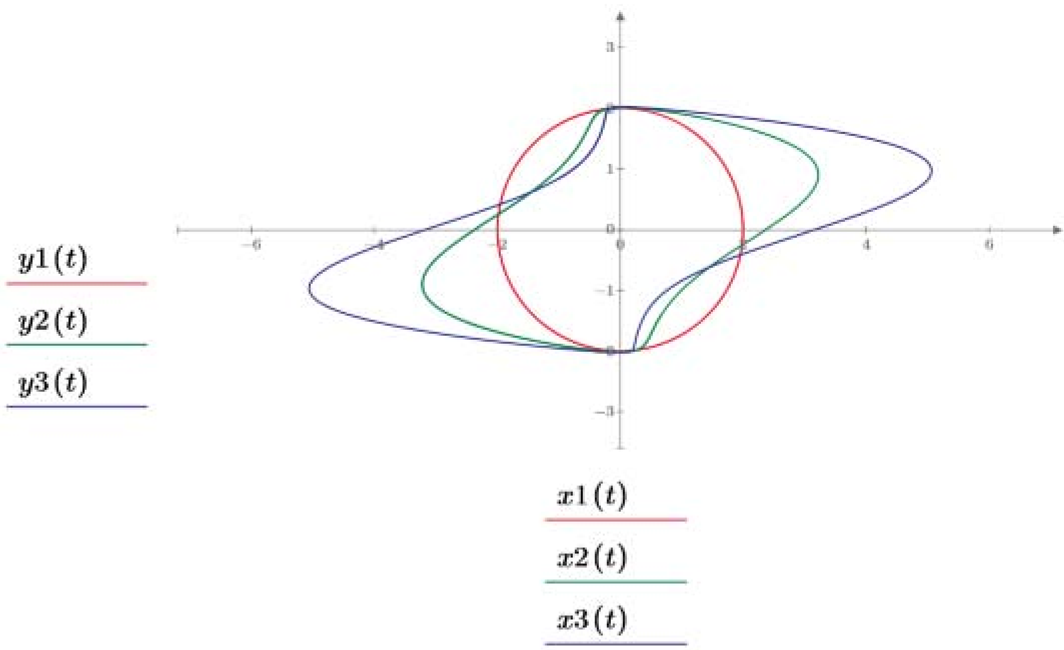
8. 여러 시스템 매개변수에 대한 해를 추출합니다.

$$\begin{bmatrix} x1 \\ y1 \end{bmatrix} := F(0)$$

$$\begin{bmatrix} x2 \\ y2 \end{bmatrix} := F(1.5)$$

$$\begin{bmatrix} x3 \\ y3 \end{bmatrix} := F(3)$$

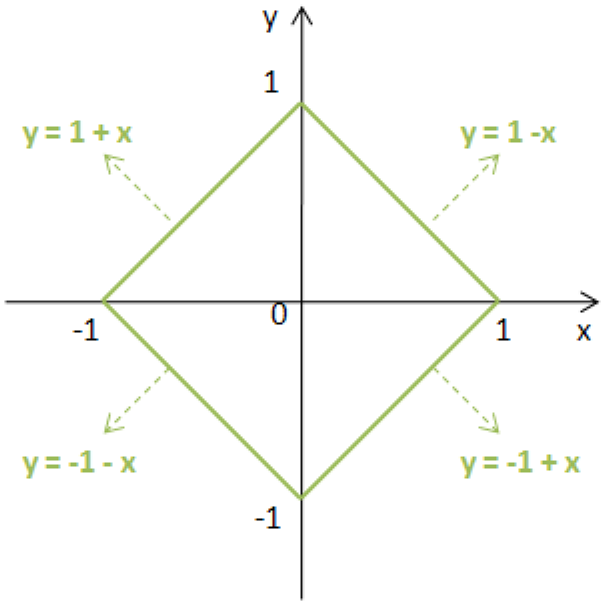
9. 해를 도표화합니다.



작업 3–5로 이동합니다.

작업 3–5: 야코비 사용

PTC Mathcad의 일부 ODE 풀이 시스템에 야코비가 사용됩니다. 야코비를 사용하면 여러 적분에 대한 변수를 변환할 수 있습니다. 함수를 적분할 다음과 같은 영역을 고려합니다. 각 경계에 대한 방정식도 표시되어 있습니다.



1. 적분할 함수를 정의합니다.

   $f(x,y) := 2\,x^3 - 5\,y^2$

2. 영역에 대해 함수를 적분합니다. 적분을 두 부분으로 나누어야 합니다. 먼저 x-y 평면의 왼쪽을 적분한 다음 오른쪽을 적분합니다.

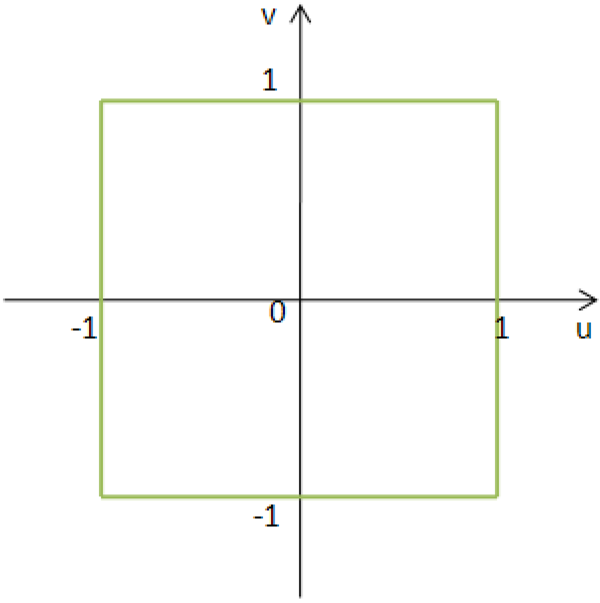
   $$\int_{-1}^{0}\int_{-1-x}^{1+x} f(x,y)\,dy\,dx + \int_{0}^{1}\int_{-1+x}^{1-x} f(x,y)\,dy\,dx = -1.667$$

   평면을 변환하는 새 변수를 도입하여 적분을 단순화할 수 있습니다.

   $u = x + y$

   $v = x - y$

   이 새 변수에 대한 적분 영역의 경계는 축과 평행합니다.



3. *u* 및 *v*를 사용하여 *x* 및 *y*를 정의합니다.

$$x(u,v) := \frac{u+v}{2}$$

$$y(u,v) := \frac{u-v}{2}$$

다중 적분에 대한 변수를 변환할 경우 야코비를 계산하여 적분의 배율을 조정해야 합니다.

4. 벡터 함수 *F(u, v)*를 정의합니다.

$$F(u,v) := \begin{bmatrix} x(u,v) \\ y(u,v) \end{bmatrix}$$

5. *a* 및 *b*에 대해 야코비 행렬을 계산합니다.

$$a := 0$$

$$b := 0$$

$$J := Jacob\left(F(a,b), \begin{bmatrix} a \\ b \end{bmatrix}\right) = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$$

6. *J*의 행렬식인 야코비를 계산합니다. 행렬식 연산자를 삽입합니다.

$$D := \|J\| = -0.5$$

7. 새 좌표를 사용하여 *f* 함수를 다시 작성합니다.

$$g(u,v) := f(x(u,v), y(u,v))$$

8. 야코비의 절대값을 사용하여 적분의 배율을 조정하고 결과를 계산합니다.

$$\int_{-1}^{1} \int_{-1}^{1} g(u,v) \cdot |D| \, du \, dv = -1.667$$

새 변수를 사용하면 적분 하나만 사용해도 함수를 적분할 수 있습니다.

실습

자습서를 마치기 전에 공중에 던진 물체가 최고점에 도달하는 시간을 구합니다. 미분 방정식 *x″ = -9.8*과 초기 조건 *x(0) = 2* 및 *x′(0) = 3*을 사용하여 풀이 구간을 설정합니다. 첫 번째 풀이 구간으로 구한 함수를 최적화하는 두 번째 풀이 구간을 설정합니다.

0 < t < 1 범위에서 첫 번째 풀이 구간으로 구한 함수를 도표화하여 해답을 확인할 수 있습니다. 계산에서 호환되는 한, 단위는 언제라도 추가할 수 있습니다.

축하합니다! 풀이 자습서를 완료했습니다.