한국수학교육학회지〈수학교육〉
1992. 8. 제 31 권 제 3 호 103-116

Journal of the Korea Society of Mathematical Education
June. 1992: Vol. 31, No. 1, 103-116

수학적 사고력을 발달시키기 위한 인지 과학 기술의 사용

황 혜정 (한국교육개발원)

수학 교육에서 수학적 사고의 중심적 역할은 더욱 강조되고 있다 (Pea, 1987; 류 희찬, 1992). 인지 심리학자들을 비롯하여 수학 분야에 관련된 심리학자들은 '사람 들이 수학적으로 어떻게 사고하는 가'에 대한 이해에 관심을 두고 있으며, 최근 들 어, 연구자들은 '어떻게 하면 좋은 사 고력이 습득되고 유지되는가'를 조사하기 시작 하였다. 특히, 수학적 인지에 관한 최근 연구 동향은, 문제해결 활동을 통하여 사 고력을 향상시키는데 있어서 수학이 중추 역활을 한다고 제시하며, 틀에 박힌 조작 보다는 해석적인 작업(work)을 전개하여 그 의미를 찾는 것을 중요한 과제로 삼고 있다 (Greer, 1981 과 Pea, 1987). Schoenfeld(1985)는 "수학에서 새로운 문제를 다룰 수 있는 능력이 유동적이고 효율적이고 지략이 풍부할 때, 수학적으로 어떻게 사고하는가를 이해한다" 라고 하였다 (p. 2).

수학적 상황을 전개하는 과정에 대한 모델이 다음에 제시된다. 이 모델에서, 학 생은 우선 문제 상황을 해석하여 수학적 용어로 바꾼다 ; 학생들이 '수학화'해가는 과정을 습득하는 것은 중요하다. 그때, 그는 구체적인 수학적 해결 방법에 접근하 기 위하여 사전 지식과 경험을 토대로 수학적 표현에 대해 생각하게 되는데, 이 과 정에서 과학 기기들은 수학적 표현과 해결 방법 사이에 수학적 사고를 발전시키기 위한 도구로서 사용될 수 있다 (그림 1 참고). 결국, 이러한 수학적인 해결 방법은 실생활의 장면에 적용되거나 표현되고 그는 다시 새로운 문제 상황을 만나게 된다.

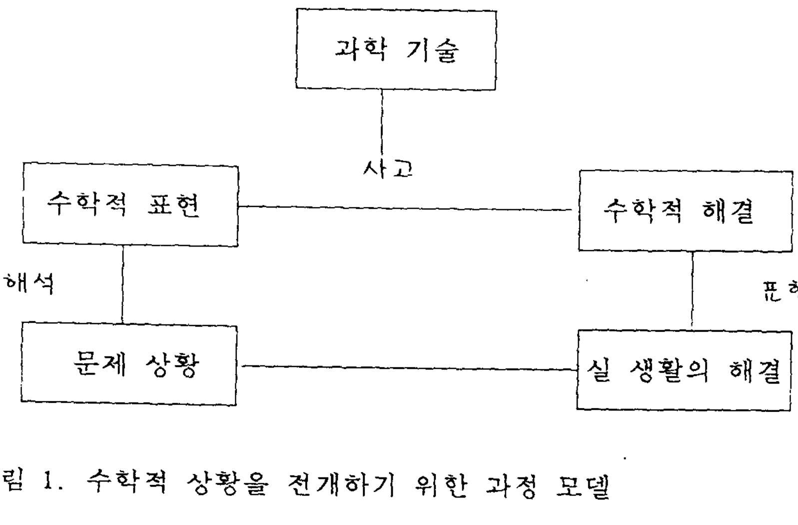


그림 1. 수학적 상황을 전개하기 위한 과정 모델

성공적인 수학 학습자들은 보다 인지적인 행동을 취하는데, 가령 수학을 학습하 는 동안, 질

차에 대한 그 자신의 이해를 검토하고 일관성을 제어하며 사전 지식을 새로운 자료와 관련지으려고 애쓴다. 또한, 그들은 문제에 도전하기 위한, 그리고 해결 가능성이 있는 부수적 문제들을 생성하기 위한, 대안적인 전략을 모색한다 (Resnick, 1987, pp. 14-15). 이와 같은 인지적 과정은 광범위한 수학적 상황에 수 학적 사고 능력을 향상시키도록 돕는다. 학생들에게 기본적인 인지 능력들을 사용 하도록 지도함으로써 학생들의 수학적 사고 능력을 향상시키는 것은 중요하다. 이 러한 인지 능력들에는 '정의하기', '서술하기'(분석, 개념과 그림 표상, 대안 표상 의 일반화로 조직화됨), '비교하기', '정당화하기', '요약하기'가 포함된다 (Swing, et. al., 1988).

한편, Pea(1987)는 학생들의 수학적 사고를 향상시키기 위한 두 가지 형태의 기 능을 제시하고 있는데, 첫째 학생들이 수학적으로 사고하도록 고무하기 위한 목적 을 조장하는 '목적 기능'(purpose functions)과 둘째 일단 그들이 수학적으로 사고 하기 시작하면, 이를 위하여 필요한 과정을 증진시키는 '과정 기능'(process functions)을 둔다 (p. 89와 p. 100). 이때, 수학적 사고를 위한 '과정', 즉 정신 적 활동들은, 위에서 언급된 바 있는, 기본적인 인지 능력들과 일맥상통한다. 또 한, 목적 기능과 과정 기능은 모든 인지 과학 기술에 적용 가능하며, 특히 과정 기능은 과학 기술에 의해 지원됨으로써 더욱 활성화될 수 있다. 따라서, '서술하기', '비교하기', '정당화하기' 등과 같은 인지 능력들은 수학적 사고를 발달시키기 위 한, 특히 수학적 사고에서 과학 기기들의 사용에 의해 지원되는 과정의 주된 구성 요소로서 강조되어야 한다 (그림 2 참고).

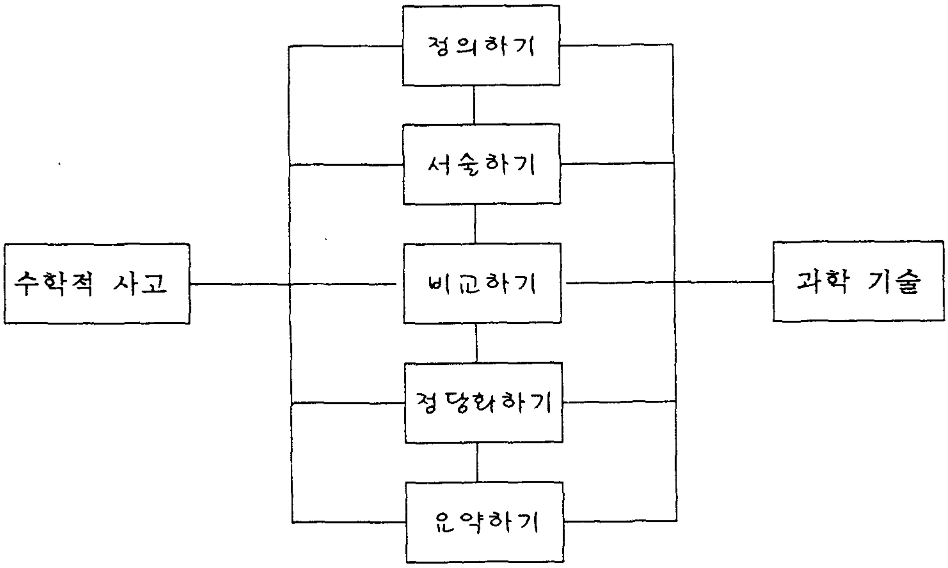


그림 2. 수학적 사고 발달과 과정 기능 증진을 위한 주요 인지 능력

인지 과학 기술(cognitive technology)은, 생각할 때, 학습할 때, 문제해결 활동 시에 정신(mind)의 제한을 초월하도록 돕는 매개체인데, 이것은 여러 가지 인간적 사고의 기능, 이해력, 과거의 지적 성취 등에 특출한 결과를 가져왔다. 이러한 기 술에는 모든 기호 체계(symbol systems), 즉 쓰기 체계, 논리, 수학적 기수법 체계, 모델, 이론, 필름과 다른 그림 매체, 그리고 현재 기호적 컴퓨터 언어가 포함되며, 특히 주된 인지적 도구로서는 문자 언어와 수학적 기수

법 체계(예를 들어, 대수, 미 적분학, 수의 상징)를 들 수 있다 (Pea, 1987, p. 91). 인지 과학 기술의 일반적인 특징은 "...사고의 중간 결과를 밖으로 드러내어 그것이 분석, 반성, 토론될 수 있 도록 한다. 주의와 기억의 왜곡과 제한에 의한 일시적이면서도 개인적인 사고 과정 들은, 그 자체로 분석의 대상이 될 수 있는 자료 기록을 공급하면서 지속하는, 전달 가능한 매개체에 포획되고(captured) 구현된다" (Pea, 1987, p. 91).

컴퓨터는 기호를 저장하고 역학적으로 조작하기 위한 보편적인 기계이다. 이와 같이, 컴퓨터는 수 뿐만 아니라 기호- 인간 사고의 기본적 유통- 를 가지고 작용한 다는 점에서 수학적으로 사고하도록 학습하기 위한 유력한(potent) 인지적 과학 기 술의 형태라 할 수 있다 (Pea, 1987, p. 92). 컴퓨터 과학 기술은 학생들이 구체적 인 수학적 사고력에서 추상적인 것으로 변화하고, 수학적 개념이나 아이디어를 조사 하여 발견할 수 있도록 돕는다 (NCTM, 1988, p. 21). 컴퓨터 사용은 학생들이 고립 된 저차원적 수준의 수학적 기술에 대한 훈련과 실습으로부터 탈피해서 나선식의 의 미있는 학습이 진행될 수 있도록 돕는다. Steen(1987)은 컴퓨터가 무엇이 가능한지 또한 무엇이 가치있는지에 대한 우리의 견해를 바꾸고, 더 나아가 수학에서 '사고하는 것'과 '작업하는 것' 사이의 균형을 재정립할 것이라고 주장한다 (p. 232). 따 라서, 컴퓨터 사용에 의하여 수학적 사고는 어떠한 영향을 받는지, 컴퓨터의 존재에 의하여 학생들은 어떻게 보다 더 수학적으로 사고할 수 있는지에 관해 관심을 둘 필 요가 있는데, 수학에서의 컴퓨터 사용은 일반적으로 세 가지 형태로 나눠질 수 있다 (Hwang, 1991, p. 17) : 수학적 소프트웨어 프로그램(prerecorded software), 프로 그래밍(teacher or student written programs), 기호적 소프트웨어 패키지 (symbolic software packages).

일반적으로, 문제 해결을 위한 소프트웨어 프로그램을 이용할 때 학생들은 정확 히 그리고 직접적으로 컴퓨터 화면에 반응해야 하기 때문에, 이런 경우에 있어서는 학생들이 그 프로그램의 지시에 따라야 하고 통제되는 경향이 있다. 이러한 프로그 램들은 학생들의 다양한 능력의 성취 정도를 평가하기 위한 검토자로서 적당할 것이 다. 그러나, 학생들이 의미있는 해석을 통하여 주어진 문제를 수학적 상황으로 이 해하고, 이를 전개, 분석하여 문제 해결에 접근해야 한다는 점을 고려할 때, 그러한 프로그램들은 학생들이 사고 과정 보다는 정확한 반응 중심의 성공적인 문제 해결자 가 되도록 돕는다고 볼 수 있다. 한편, Al-Orainy(1984)가 지적하기를, "컴퓨터 과 학 교육에 관련된 사람들 사이에 공통적인 의견이 있는데, 프로그래밍은 [그것을]필요로 하는 절차적 계획을 촉진하기 때문에, 이것은 좋은 문제해결력의 성장을 향상 시킨다"고 한다 (p. 17). 그러나, Pea(1987, p. 99)에 따르면, 프로그래밍 언어 사 용은 학생들이 수학적으로 사고하는 것을 돕긴 하지만, 프로그래밍 언어 배열 (syntax)에 관한 훈련과 실습, 또는 어떤 프로그램을 작성하기 위한 추상적인 연습 은 학생들의 수학적 사고를 과히 요구하지 않는다.

반면에, 기호적 소프트웨어 패키지(예를 들어, TK! Solver, Maple, muMath, Mathematica 등)의 적절한 사용은 학생들이 수학적 아이디어를 일반화하거나 조직화 함으로써 수학적 사고를 전개하고 해결 방법에 접근함으로써 일련의 사고 과정을 통 합하도록 돕는다. 또한, 이것은 컴퓨터 절차들(computer procedures)을 구성함으로 서 중요한 대수적(수적, 기호적, 연산적)이고 기하학적인 학습을 성취하도록 도우 며, 학생들에게 결과를 찾아내고 그들 나름대로 그 해

결책을 확실히 밝히게 하는 기 회를 제공한다. 특히, 눈으로 보고 탐구할 수 있는(visualization 과 exploration) 장점을 가진 기호적 소프트웨어 패키지의 사용은 학생들이 더 나은 수학적 사고력을 소유하도록 돕는 도구로서 권장된다 (Pea, 1987). 결론적으로, 학생들의 수학적 사 고를 발전시키고 더 나아가 그들의 학업 성취도를 높이기 위하여, 수학적(기호적) 체계 형태의 소프트웨어 과학 기술의 사용에 관한 구체적인 방법이 모색되어야 할 것이다. 따라서, 본고는 주요 인지 능력들에 중점을 두어 컴퓨터(symbolic software packages: Mathematica)의 사용과 이해에 관해 간단히 밝히고자, 다음에 예문을 제시한다.

예)

> 어떤 용기 제조 회사는 과일 쥬스를 담을 원기둥 모양의 깡통을 만들려고 한다. 깡통 하나의 부피는 0.946 리터이다. 생산 비용을 최대한 도로 줄이기 위하여, 이 회사에서는, 가능한 한 최소량의 재료를 사용하여 깡통을 만들려고 한다. 이 때 깡통의 크기는 얼마이겠는가? (*Curriculum and evaluation standards for school mathematics*, p. 134)

정의하기 :

1 liters = 1000 ml, 1 ml = 1 cm3 임을 안다. 따라서, 깡통의 부피가 cm3 단위로 표현될 수 있음을 안다 ; 깡통의 부피는 946 cm3 이다. 작은량의 재 료를 사용할수록 깡통의 표면적이 작아짐을 알고, 문제 조건을 만족시키기 위 해서는 깡통의 표면적이 최소값을 취해야 함을 안다. 깡통의 높이(height)와 표면적(surface)의 공식을 알아, 이들을 Mathematica를 사용하여 정의해 본다 (Mathematica In[1]과 Out[1], In[2]과 Out[2] 참고).

서술하기 :

함수 surface에 대한 그래프의 개형(반경; x축, 표면적; y축)을 알아 본다. 이를 위하여, Mathematica의 Plot 명령문을 사용한다.

비교하기 :

이때(Plot 명령문 사용시), x축(반경, r)의 범위 간격을 적절히 조절, 변경 (시행 착오 전략)하여 적절한 그래프의 개형을 얻도록 한다 (In[3] 참고).

서술하기 :

반경과 표면적은 0 보다 작은 값(0도 포함)을 포함하지 않으므로 이들 값의 무의미함을 알고, 출력된 그래프(Out[3])에서 1사분면에 나타나 있는 그래프 만이 주어진 문제 상황에 필요함을 안다. 표면적의 최소값의 위치를 보다 정 확히 파악한다. 이를 위하여 Mathematica의 Plot 명령문을 사용한다.

비교하기 :

이때, 반경(x축)의 간격을 적절히 좁혀, 표면적에 대한 그래프를 얻도록 한다 (In[4] 참고).

서술하기 :

출력된 그래프(Out[4])를 통하여, 반경이 5 정도일 때 표면적이 최소가 됨을 안다.

비교하기 :

표면적의 최소값을 만족시키기 위한 보다 정확한 반경의 값을 얻는다. 이를 위하여, Mathematica의 Table 명령문을 사용한다 (In[5] 참고).

서술하기 :

출력된 표(Out[5])를 통하여 반경이 5 와 6 사이에 존재할 때 표면적이 최소 가 됨을 안다. 5 와 6 사이의 반경의 값에 대응하는 표면적의 값을 알아본 다. 이를 위하여, Table 명령어를 다시 사용한다 (In[6] 참고). 문제 조건 을 만족하는 표면적의 값을 구할 때까지 위의 활동(Table 명령어 사용)이 반 복되도록 한다 . 출력된 표(Out[6])를 통하여 반경이 5.3 과 5.4 사이에 존재할 때 표면적이 최소가 됨을 안다. 5.3 과 5.4 사이의 반경의 값에 대응하 는 표면적의 값을 알아본다. 이를 위하여, Table 명령문을 사용하고 (In[7] 참고), 출력된 표(Out[7])를 통하여 반경이 5.32 일 때 표면적이 최소가 됨을 발견한다.

정당화하기 :

구한 답이 맞았는지를 검토한다. 이를 위하여 Mathematica의 FindMinimum 명 령문을 사용해 본다 (In[8] 참고). 이 결과에 따르면 (Out[8] 참고), 반경이 5.3199 일때 표면적의 값이 최소가 되므로, 위에서 구한 반경의 값, 5.32는 이 문제 조건에 타당한 근사값임을 밝힌다. 이를 위하여 Mathematica의 명령 문들을 활용하여 본다 . Table 명령문을 사용하면 (In[9] 참고), 5.317에서 5.322 사이의 값에서 모 두 표면적은 같은 최소값을 나타내지만 (Out[9] 참고), Plot 명령문을 사용하 여 이 값들을 그래프로 나타내보면 (In[10] 참고), 5.32 정도의 값을 취하는 반경에서 최소의 표면적을 나타낸다 (Out[10] 참고) .

요약하기 :

깡통의 반경이 약 5.3 cm 이고 표면적이 약 533.5cm^2 일 때, 이 문제의 조건 을 만족시킬 수 있음을 안다. 즉, 최소한의 비용으로 깡통을 만들기 위해서 는, 깡통 하나의 재료 면적은 약 533.5cm^2 가 됨을 안다. 참고로, 깡통의 높이와 표면적의 관계가 어떠한 지를 알아 본다 . 이를 위하 여 Table과 ListPlot 명령문을 사용해 본다 (In[11], In[12], In[13] 참고). 이 결과에 따라 (Out[11], Out[12], Out[13] 참고), 반경에 따른 깡통의 높이 (그래프의 x축)와 표면적(그래프의 y축)은 비례하지 않음을 안다.

Mathematica, "그것은 프로그래밍 언어, 수학 도구, 교수 도구, 유용물, 기호적 조작적 기계...이다" (Burgard, 1989, p. 90). Mathematica는 '컴퓨터에 의해 수학 을 하기 위한 체계'(system for doing mathematics)라 불리우는데, 이를 사용함으로 써 학생들은 수 계산 및 기호에 대한 계산을 수행하고, 그래픽 표현을 산출하고, 그 들 자신이 함수를 정의해 보고, 프로그램을 작성해 보고, 수학적 지식을 표현 해 볼 수 있다 (Wolfram, 1988). 특히, Mathematica를 포함하여, 기호적 소프트웨어 패키 지(symbolic software package)는 학습에 관한 인지적 방법을 위하여 효과적인 교수 환경이 될 수 있다. 다시 말하면, 기호적 소프트웨어의 유동성(flexibility)과 기 능성(functionality)은 학생들에게 그들의 수학적 사고를 향상시키기 위한 최상의 조건-적절한 절차를 조사하고, 답을 발견하고, 이러한 답의 타당성을 판단하는 것 -을

부여한다. 따라서, 학생들은 이러한 소프트웨어를 활용하여 문제를 해결함으 로써 보다 융통성 있고 의미있는 수학적 사고를 전개할 수 있으며, 또한 이러한 경험 을 통해 보다 나은 수학적 사고력을 형성하기 위하여 과연 컴퓨터가 어떻게 사용될 수 있는지에 관한 아이디어를 가질 수 있다.

기본적 사실을 회상하거나 단순한 암기가 아닌, 사고력을 지도하는 중요한 이유 는 개념적 발달, 수학에 대한 믿음, 심화된 학습을 위한 것이다 (NCTM, 1989). 수 학적 상황을 전개해 나아갈 때, 사고력은 지각(해석하는 것)과 행위(표현하는 것) 사이의 관계를 구성하게 되는데, 이때 과학 기술은 이러한 관계를 자극하는 수단 (ways)으로 간주되어야 할 것이다 (그림 1 참고). 따라서, 강한(powerful) 사고력 은 점차적으로 '인지적' 과학 기술의 활용과 이에 따른 세련된 '개념적' 사고력의 전개와 더불어 형성될 것이다. Resnick(1987)에 따르면, "...학교는 모든 학생들에 게 사고력을 기르기 위한 방법들을 모색하고 있다. 어떠한 교육 제도도 여지껏, 명사(elite)가 아닌 모든이가 유능한 사고력을 지닌 자가 될 수 있다는 가정을 전제로 하지 않아 왔었다. 우리는 이러한 새로운 도전을 창의적이고 바로 요구하는 교육 개혁을 위한 권유로 받아들여야 한다.... 개혁을 위한 도전은, 인지적 연구가 사고 와 학습의 본질에 대한 중요한 재개념화를 공급하는 시기에 이른다..." (p. 45). 결론적으로, 수학 교육에서 학생들의 수학적 사고력에 대한 효과적 신장을 위한 노 력으로, 우선 새로운 과학 기술들을 사용하여 수학을 학습할 시기임을 인식하고 인 지 과학 기술의 사용에 관한 연구가 활발히 진행되어야 할 것이다.

참 고 문 헌

1. Al-Orainy, A. S., *The effect of computer literacy on problem solving in mathematics: an analysis of NAEP data (doctoral dissertation, University of Denver, 1984). Dissertation Abstracts International, 45, 3085-A.*, 1984.
2. Greer, B., *Cognitive psychology and mathematical thinking.*, vol. 1, For the Learning of Mathematics,, 1981, pp. 19–24..
3. Hwang, H. J., *Experiences of middle school mathematics teachers in Korea with materials involving calculator and microcomputer activities: Three case studies.*, Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign., 1991.
4. National Council of Teachers of Mathematics., *The use of computers in the learning and teaching of mathematics.*, Unpublished manuscript., 1988.
5. National Council of Teachers of Mathematics., *Curriculum and evaluation standards for school mathematics.*, Reston, VA: Author., 1989.
6. Pea, R. D., *Cognitive technologies for mathematics education. In A. H. Schoenfeld(Ed.), Cognitive science and mathematics education (pp. 89-122).*, Hillsdale, NJ: Lawrence Erlbaum Associates., 1987.
7. Resnick, L. B., *Education and Learning to Think. Washington, DC:*, National Academy Press. 1987.
8. Schoenfeld, A., *Mathematical Problem Solving.*, New York: Academic Press., 1985.
9. Steen, L. A., *Who still does math with paper and pencil? Calculus for a New Century (pp 231-232).*, Washington, DC: Mathematical Association of America., 1988.

10. Swing, S. R., Stoiber, K. C., & Peterson, P. L., *Thinking skills versus learning time: effects of alternative classroom-based interventions on students' mathematics problem solving. Cognition and Instruction, 5(2), 123-191.*, 1988.

11. Wolfram, S., *Mathematica.*, Menlo Park, CA: Addison-Wesley., 1988.

12. 류희찬., 수학과 교육과정 개정 방향에 대한 소고., 청람수학교육, 2,, 1992, pp. 59-70..

주.

1) 이러한 반복적 활동은 접근 방법('Approaching method'; Hwang, 1991)을 사용한 것이라 할 수 있는데, 이것은 최대값, 최소값, 또는 근을 구하기 위한 것으로써 Mathematica의 Plot 명령어와 Table 명령어를 각각 또는 병행하여 연속적으로 사용 하여 그들의 값들에 접근해 나아가는 방법이다.

2) 이때, '서술하기', '비교하기' 와 같은 인지 능력들은, 위에서 이미 나타난 것 과 같이, 상호적 반복적으로 사용된다.

3) Table과 Plot 명령어 이외에도, Solve 명령어를 이용하여 해를 구하거나 또는 ReplaceAll 명령어를 이용하여 참, 거짓임을 밝혀, 이 문제 조건에 타당한 근사값들 을 알아 볼 수 있다 (이러한 활동은 본고에서 생략됨).

4) 이때도 마찬가지로, '서술하기', '비교하기' 와 같은 인지 능력들은 상호적, 반 복적으로 사용된다.'

▩ *Mathematica* INPUT AND OUTPUT

In[1]:=

```
height=946 / (Pi r^2)
```

Out[1]=

```
 946
-----
    2
Pi r
```

In[2]:=

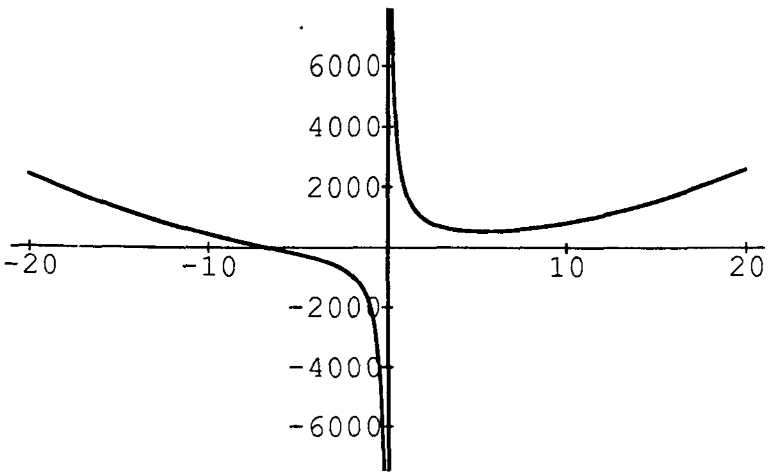
```
surface= 2 Pi r (height) + 2 (Pi r^2)
```

Out[2]=

```
1892          2
---- + 2 Pi r
 r
```

In[3]:=

```
Plot [surface, {r, -20, 20}]
```

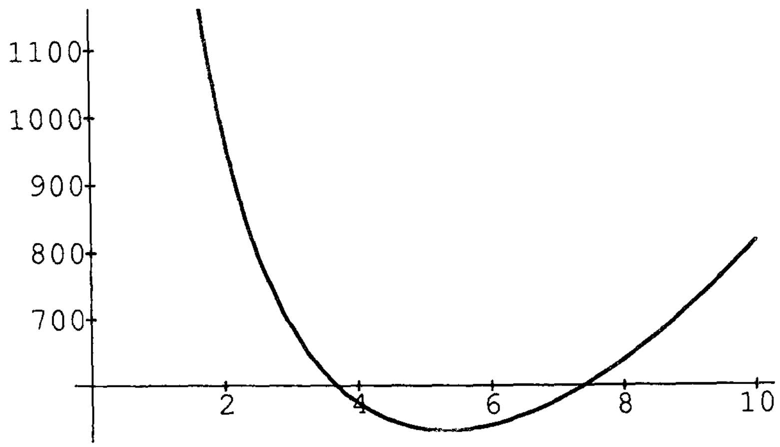


Out[3]=

```
-Graphics-
```

In[4]:=

```
Plot [surface, {r, 1, 10}]
```



Out[4]=

```
-Graphics-
```

In[5]:=

```
Table[{r, surface}, {r, 1, 10, 1}]  // N //TableForm
```

Out[5]//TableForm=

1.	1898.28
2.	971.133
3.	687.215
4.	573.531
5.	535.48
6.	541.528
7.	578.162
8.	638.624
9.	719.16
10.	817.519

In[6]:=

```
Table[{r, surface}, {r, 5, 6, 0.1}]  // N  //TableForm
```

Out[6]//TableForm=

5.	535.48
5.1	534.406
5.2	533.743
5.3	533.476
5.4	533.588
5.5	534.066
5.6	534.898
5.7	536.071
5.8	537.573
5.9	539.396
6.	541.528

In[7]:=

```
Table[{r,  surface}, {r, 5.3, 5.4, 0.01}]  // N  //TableForm
```

Out[7]//TableForm=

5.3	533.476
5.31	533.47
5.32	533.468
5.33	533.47
5.34	533.476
5.35	533.485
5.36	533.498
5.37	533.515
5.38	533.536
5.39	533.56
5.4	533.588

In[8]:=

```
FindMinimum[surface, {r, 5, 4, 6}]
```

Out[8]=

```
{533.468, {r -> 5.3199}}
```

In[9]:=

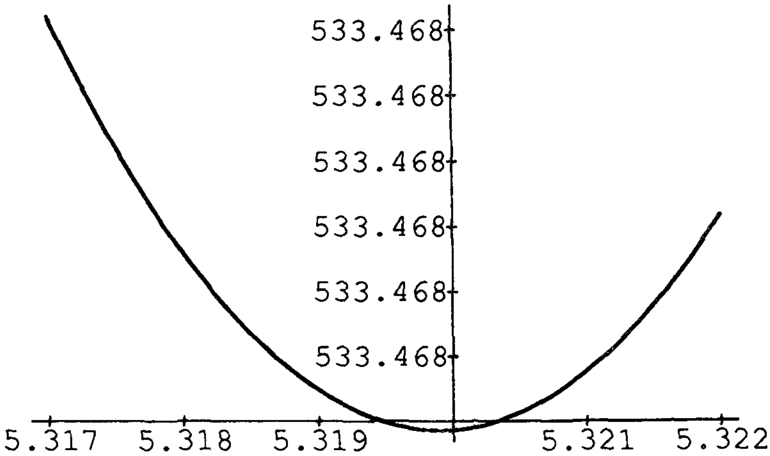
```
Table[{r,  surface}, {r, 5.315, 5.325, 0.001}]  // N  //Table
```

Out[9]//TableForm=

5.315	533.469
5.316	533.469
5.317	533.468
5.318	533.468
5.319	533.468
5.32	533.468
5.321	533.468
5.322	533.468
5.323	533.469
5.324	533.469
5.325	533.469

In[10]:=

```
Plot[surface, {r, 5.317, 5.322}]
```



Out[10]=

```
-Graphics-
```

In[11]:=

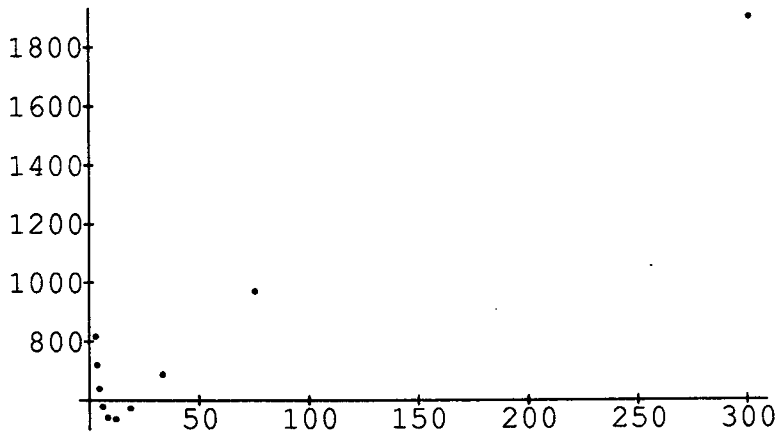
```
Table [{r, height, surface}, {r, 1, 10}] //N //TableForm
```

Out[11]//TableForm=

1.	301.121	1898.28
2.	75.2803	971.133
3.	33.4579	687.215
4.	18.8201	573.531
5.	12.0448	535.48
6.	8.36448	541.528
7.	6.14533	578.162
8.	4.70502	638.624
9.	3.71755	719.16
10.	3.01121	817.519

In[12]:=

```
ListPlot[Table[{height, surface}, {r, 1, 10, 1}]  // N]
```

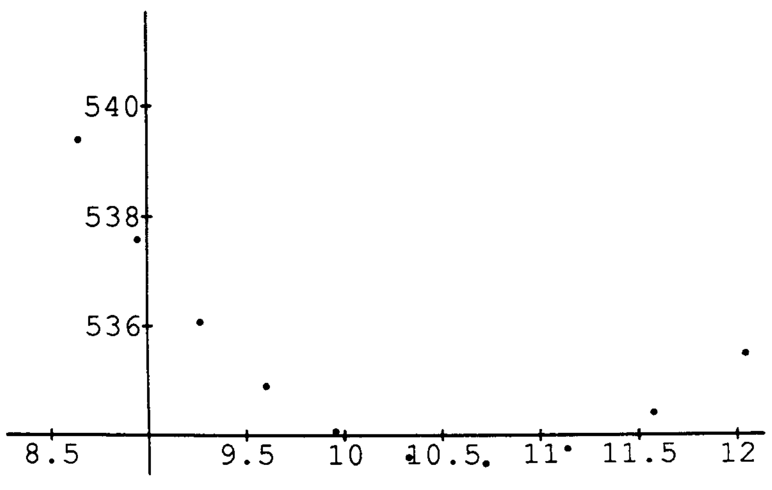


Out[12]=

```
-Graphics-
```

In[13]:=

```
ListPlot[Table[{height, surface}, {r, 5, 6, 0.1}]  // N]
```



Out[13]=

```
-Graphics-
```