Journal of Economic Perspectives—Volume 20, Number 4—Fall 2006—Pages 29–46

The Macroeconomist as Scientist and Engineer

N. Gregory Mankiw

Economists like to strike the pose of a scientist. I know, because I often do it myself. When I teach undergraduates, I very consciously describe the field of economics as a science, so no student will start the course thinking that he or she is embarking on some squishy academic endeavor. Our colleagues in the physics department across campus may find it amusing that we view them as close cousins, but we are quick to remind anyone who will listen that economists formulate theories with mathematical precision, collect huge data sets on individual and aggregate behavior, and exploit the most sophisticated statistical techniques to reach empirical judgments that are free of bias and ideology (or so we like to think).

Having recently spent two years in Washington as an economic adviser at a time when the U.S. economy was struggling to pull out of a recession, I am reminded that the subfield of macroeconomics was born not as a science but more as a type of engineering. God put macroeconomists on earth not to propose and test elegant theories but to solve practical problems. The problems He gave us, moreover, were not modest in dimension. The problem that gave birth to our field—the Great Depression of the 1930s— was an economic downturn of unprecedented scale, including incomes so depressed and unemployment so widespread that it is no exaggeration to say that the viability of the capitalist system was called into question.

This essay offers a brief history of macroeconomics, together with an evaluation of what we have learned. My premise is that the field has evolved through the efforts of two types of macroeconomists—those who understand the field as a type of engineering and those who would like it to be more of a science. Engineers are, first and foremost, problem solvers. By contrast, the goal of scientists is to under-

■ *N. Gregory Mankiw is the Robert M. Beren Professor of Economics, Harvard University, Cambridge, Massachusetts.*

stand how the world works. The research emphasis of macroeconomists has varied over time between these two motives. While the early macroeconomists were engineers trying to solve practical problems, the macroeconomists of the past several decades have been more interested in developing analytic tools and establishing theoretical principles. These tools and principles, however, have been slow to find their way into applications. As the field of macroeconomics has evolved, one recurrent theme is the interaction—sometimes productive and sometimes not—between the scientists and the engineers. The substantial disconnect between the science and engineering of macroeconomics should be a humbling fact for all of us working in the field.

To avoid any confusion, I should say at the outset that the story I tell is not one of good guys and bad guys. Neither scientists nor engineers have a claim to greater virtue. The story is also not one of deep thinkers and simple-minded plumbers. Science professors are typically no better at solving engineering problems than engineering professors are at solving scientific problems. In both fields, cutting-edge problems are hard problems, as well as intellectually challenging ones. Just as the world needs both scientists and engineers, it needs macroeconomists of both mindsets. But I believe that the discipline would advance more smoothly and fruitfully if macroeconomists always kept in mind that their field has a dual role.

The Keynesian Revolution

The word "macroeconomics" first appears in the scholarly literature in the 1940s. To be sure, the topics of macroeconomics—inflation, unemployment, economic growth, the business cycle, and monetary and fiscal policy—have long intrigued economists. In the eighteenth century, for example, David Hume (1752) wrote about the short-run and long-run effects of monetary injections; at many points, his analysis looks remarkably similar to what one might see from a modern monetary economist or central banker. In 1927, Arthur Pigou published a book titled *Industrial Fluctuations* that attempted to explain the business cycle. Nonetheless, the field of macroeconomics as a distinct and active area of inquiry arose in the shadow of the Great Depression. There is nothing like a crisis to focus the mind.

The Great Depression had a profound impact on those who lived through it. In 1933, the U.S. unemployment rate reached 25 percent, and real GDP was 31 percent below its 1929 level. All subsequent fluctuations in the U.S. economy have been ripples on a calm sea compared to this tsunami. Autobiographical essays by prominent economists of this era, such as Lawrence Klein, Franco Modigliani, Paul Samuelson, Robert Solow, and James Tobin, confirm that the Depression was a key motivating event in their careers (Breit and Hirsch, 2004).

The General Theory of John Maynard Keynes was the focal point in professional discussions about how to understand these developments. All five of these Nobel laureates confirm this from first-hand experience. Tobin reports the following reaction from Harvard, where he was a student in the late 1930s and early 1940s: "The senior faculty was mostly hostile . . . . The younger faculty and the graduate student teaching fellows were enthusiastic about Keynes's book." As is often the case, the young had greater foresight than the old about the impact of the new ideas. Keynes tied with Alfred Marshall as the most frequently cited economist in economic journals in the 1930s and was the second most cited in the 1940s, after Hicks (Quandt, 1976). This influence persisted for many years. Keynes ranked number 14 in citations for the period from 1966 to 1986, even though he died two decades before the time period began (Garfield, 1990).

The Keynesian revolution influenced not only economic research but also pedagogy. Samuelson's classic textbook *Economics* was first published in 1948, and its organization reflected his perception of what the profession had to offer to the lay reader. Supply and demand, which today are at the heart of how we teach economics to freshmen, were not introduced until page 447 of the 608-page book. Macroeconomics came first, including such concepts as the fiscal-policy multiplier and the paradox of thrift. Samuelson wrote (on p. 253), "Although much of this analysis is due to an English economist, John Maynard Keynes, . . . today its broad fundamentals are increasingly accepted by economists of all schools of thought."

When a modern economist reads *The General Theory*, the experience is both exhilarating and frustrating. On the one hand, the book is the work of a great mind being applied to a social problem whose currency and enormity cannot be questioned. On the other hand, although the book is extensive in its analysis, it somehow seems incomplete as a matter of logic. Too many threads are left hanging. The reader keeps asking, what, precisely, is the economic model that ties together all the pieces?

Soon after Keynes published *The General Theory*, a generation of macroeconomists worked to answer this question by turning his grand vision into a simpler, more concrete model. One of the first and most influential attempts was the IS–LM model proposed by the 33-year-old John Hicks (1937). The 26-year-old Franco Modigliani (1944) then extended and explained the model more fully. To this day, the IS–LM model remains the interpretation of Keynes offered in the most widely used intermediate-level macroeconomics textbooks. Some Keynesian critics of the IS–LM model complain that it oversimplifies the economic vision offered by Keynes in *The General Theory*. To some extent, this may well be true. The whole point of the model was to simplify a line of argument that was otherwise hard to follow. The line between simplifying and oversimplifying is often far from clear.

While theorists such as Hicks and Modigliani were developing Keynesian models suitable for the classroom blackboard, econometricians such as Klein were working on more applied models that could be brought to the data and used for policy analysis. Over time, in the hope of becoming more realistic, the models became larger and eventually included hundreds of variables and equations. By the 1960s, there were many competing models, each based on the input of prominent Keynesians of the day, such as the Wharton model associated with Klein, the DRI (Data Resources, Inc.) model associated with Otto Eckstein, and the MPS (MIT–Penn–Social Science Research Council) model associated with Albert Ando and Modigliani. These models were widely used for forecasting and policy analysis. The MPS model was maintained by the Federal Reserve for many years and would become the precursor to the FRB/US model, which is still maintained and used by Fed staff.

Although these models differed in detail, their similarities were more striking than their differences. They all had an essentially Keynesian structure. In the back of each model builder's mind was the same model taught to undergraduates today: an IS curve relating financial conditions and fiscal policy to the components of GDP, an LM curve that determined interest rates as the price that equilibrates the supply and demand for money, and some kind of Phillips curve that describes how the price level responds over time to changes in the economy.

As a matter of science, *The General Theory* was a remarkable success. The revolution that it inspired attracted many of the best young minds of its day. Their prodigious output offered a new way to understand short-run economic fluctuations. Reflecting on these events, Samuelson (1988) offered a succinct summary: "The Keynesian revolution was the most significant event in 20th-century economic science." This sentiment is shared by many economists of his generation.

Yet the Keynesian revolution cannot be understood merely as a scientific advance. To a large extent, Keynes and the Keynesian model builders had the perspective of engineers. They were motivated by problems in the real world, and once they developed their theories, they were eager to put them into practice. Until his death in 1946, Keynes himself was heavily involved in offering policy advice. So, too, were the early American Keynesians. Tobin, Solow, and Eckstein all took time away from their academic pursuits during the 1960s to work at the Council of Economic Advisers. The Kennedy tax cut, eventually passed in 1964, was in many ways the direct result of the emerging Keynesian consensus and the models that embodied it.

The New Classicals

By the late 1960s, cracks in the Keynesian consensus were starting to appear. Those cracks would grow into fissures, which would eventually crumble the macroeconomic consensus and undermine confidence in the mainstream econometric models. In its place, a more classical view of the economy would reemerge.

The first wave of new classical economics was monetarism, and its most notable proponent was Milton Friedman. Friedman's (1957) early work on the permanent income hypothesis was not directly about money or the business cycle, but it certainly had implications for business cycle theory. It was, in part, an attack on the Keynesian consumption function, which provided the foundation for the fiscal policy multipliers that were central to Keynesian theory and policy prescriptions. If the marginal propensity to consume out of transitory income is small, as Friedman's theory suggested, then fiscal policy would have a much smaller impact on equilibrium income than many Keynesians believed.

Friedman and Schwartz's (1963) *Monetary History of the United States* was more directly concerned with the business cycle and it, too, undermined the Keynesian consensus. Most Keynesians viewed the economy as inherently volatile, constantly buffeted by the shifting "animal spirits" of investors. Friedman and Schwartz suggested that economic instability should be traced not to private actors, but rather to inept monetary policy. The implication was that policymakers should be satisfied if they do no harm by following simple policy rules. Although Friedman's proposed rule of steady growth in monetary aggregates has few adherents today, it was an early precursor to the inflation-targeting regimes now in effect in many of the world's central banks.

Friedman's Presidential Address to the American Economic Association in 1968, along with Phelps (1968), took aim at the weakest link in the Keynesian model: the Phillips curve trade-off between inflation and unemployment. At least since Samuelson and Solow (1960), some sort of Phillips curve had been part of the Keynesian consensus, even if not a view endorsed by Keynes himself. Samuelson and Solow (1960) understood the theoretical tenuousness of this trade-off, and their paper was filled with caveats about why the short-run and long-run trade-off could differ. But the subsequent literature forgot those caveats all too easily. The Phillips curve provided a convenient way to complete the Keynesian model, which always had trouble explaining why prices failed to equilibrate markets and how the price level adjusted over time.

Friedman argued that the trade-off between inflation and unemployment would not hold in the long run, when classical principles should apply and money should be neutral. The trade-off appeared in the data because, in the short run, inflation is often unanticipated and unanticipated inflation can lower unemployment. The particular mechanism that Friedman suggested was money illusion on the part of workers. More important for the development of macroeconomics was that Friedman put expectations on center stage.

This prepared the way for the second wave of new classical economics—the rational expectations revolution. In a series of highly influential papers, Robert Lucas extended Friedman's argument. In his "Econometric Policy Evaluation: A Critique," Lucas (1976) argued that the mainstream Keynesian models were useless for policy analysis because they failed to take expectations seriously; as a result, the estimated empirical relationships that made up these models would break down if a new policy were implemented. Lucas (1973) also proposed a business cycle theory based on the assumptions of imperfect information, rational expectations and market clearing. In this theory, monetary policy matters only to the extent to which it surprises people and confuses them about relative prices. Barro (1977) offered evidence that this model was consistent with U.S. time-series data. Sargent and Wallace (1975) pointed out a key policy implication: Because it is impossible to surprise rational people systematically, systematic monetary policy aimed at stabilizing the economy is doomed to failure.

The third wave of new classical economics was the real business cycle theories of Kydland and Prescott (1982) and Long and Plosser (1983). Like the theories of Friedman and Lucas, these were built on the assumption that prices adjust instantly to clear markets—a radical difference from Keynesian theorizing. But unlike the new classical predecessors, the real business cycle theories omitted any role of monetary policy, unanticipated or otherwise, in explaining economic fluctuations.

The emphasis switched to the role of random shocks to technology and the intertemporal substitution in consumption and leisure that these shocks induced.

As a result of the three waves of new classical economics, the field of macroeconomics became increasingly rigorous and increasingly tied to the tools of microeconomics. The real business cycle models were specific, dynamic examples of Arrow–Debreu general equilibrium theory. Indeed, this was one of their main selling points. Over time, proponents of this work have backed away from the assumption that the business cycle is driven by real as opposed to monetary forces, and they have begun to stress the methodological contributions of this work. Today, many macroeconomists coming from the new classical tradition are happy to concede to the Keynesian assumption of sticky prices, as long as this assumption is imbedded in a suitably rigorous model in which economic actors are rational and forward-looking. Because of this change in emphasis, the terminology has evolved, and this class of work now often goes by the label "dynamic stochastic general equilibrium" theory. But I am getting ahead of the story.

At the time the three new classical waves were first hitting shore in the 1970s and 1980s, one of their goals was to undermine the old Keynesian macroeconometric models both as a matter of science and as a matter of engineering. In their article "After Keynesian Macroeconomics," Lucas and Sargent (1979) wrote, "For policy, the central fact is that Keynesian policy recommendations have no sounder basis, in a scientific sense, than recommendations of non-Keynesian economists or, for that matter, noneconomists." Although Sargent and Lucas thought Keynesian engineering was based on flawed science, they knew that the new classical school (circa 1979) did not yet have a model that was ready to bring to Washington: "We consider the best currently existing equilibrium models as prototypes of better, future models which will, we hope, prove of practical use in the formulation of policy." They also ventured that such models would be available "in ten years if we get lucky." I will return later to the question of whether this prospect panned out as they had hoped.

As these quotations suggest, those engaged in the new classical movement were neither shy about their intentions nor modest about their accomplishments. Lucas offered an even more blunt assessment in a 1980 article entitled "The Death of Keynesian Economics": "One cannot find good, under-forty economists who identify themselves or their work as 'Keynesian'. Indeed, people even take offense if referred to as 'Keynesians'. At research seminars, people don't take Keynesian theorizing seriously anymore; the audience starts to whisper and giggle to one another." Yet just as Lucas was writing the eulogy for Keynesian economics, the profession was about to welcome a generation of "new Keynesians."

The New Keynesians

Economists attracted to the Keynesian approach to the business cycle have long been discomfited by the issue of microfoundations. Indeed, a 1946 article by Lawrence Klein, one of the first to use the term "macroeconomics," begins as follows: "Many of the newly constructed mathematical models of economic systems, especially business-cycle theories, are very loosely related to the behavior of individual households or firms which must form the basis of all theories of economic behavior." All modern economists are, to some degree, classical. We all teach our students about optimization, equilibrium, and market efficiency. How to reconcile these two visions of the economy—one founded on Adam Smith's invisible hand and Alfred Marshall's supply and demand curves, the other founded on Keynes's analysis of an economy suffering from insufficient aggregate demand—has been a profound, nagging question since macroeconomics began as a separate field of study.

Early Keynesians, such as Samuelson, Modigliani, and Tobin, thought they had reconciled these visions in what is sometimes called the "neoclassical–Keynesian synthesis." These economists believed that the classical theory of Smith and Marshall was right in the long run, but the invisible hand could become paralyzed in the short run described by Keynes. The time horizon mattered because some prices—most notably the price of labor—adjusted sluggishly over time. Early Keynesians believed that classical models described the equilibrium toward which the economy gradually evolved, but that Keynesian models offered the better description of the economy at any moment in time when prices were reasonably taken as predetermined.

The neoclassical–Keynesian synthesis is coherent, but it is also vague and incomplete. While the new classical economists responded to these defects by rejecting the synthesis and starting afresh, the new Keynesian economists thought there was much to preserve. Their goal was to use the tools of microeconomics to give greater precision to the uneasy compromise reached by early Keynesians. The neoclassical–Keynesian synthesis was like a house built in the 1940s: The new classicals looked at its outdated systems and concluded it was a tear-down, while the new Keynesians admired the old-world craftsmanship and embraced it as an opportunity for a major rehab.

The first wave of research that can rightly be called "new Keynesian" is the work on general disequilibrium (Barro and Grossman, 1971; Malinvaud, 1977). These theories aimed to use the tools of general equilibrium analysis to understand the allocation of resources that results when markets do not clear. Wages and prices were taken as given. The focus was on how the failure of one market to clear influences supply and demand in related markets. According to these theories, the economy can find itself in one of several regimes, depending on which markets are experiencing excess supply and which are experiencing excess demand. The most interesting regime—in the sense of corresponding best to what we observe during economic downturns—is the so-called "Keynesian" regime in which both the goods market and the labor market are exhibiting excess supply. In the Keynesian regime, unemployment arises because labor demand is too low to ensure full employment at prevailing wages; the demand for labor is low because firms cannot sell all they want at prevailing prices; and the demand for firms' output is inadequate because many customers are unemployed. Recessions and depressions result from a vicious circle of insufficient demand, and a stimulus to demand can have multiplier effects.

The second wave of new Keynesian research aimed to explore how the concept of rational expectations could be used in models without the assumption of market clearing. To some extent, this work was responding to Sargent and Wallace's (1975) conclusion of monetary policy irrelevance by showing how systematic monetary policy could potentially stabilize the economy, despite rational expectations (Fischer, 1977). To some extent, it was motivated by a desire to find an empirically realistic model of inflation dynamics (Taylor, 1980). The Achilles heel of this work was that it assumed a form of labor contracts that, while perhaps justifiable on empirical grounds, was hard to square with microeconomic principles.

Because so much of the Keynesian tradition was based on the premise that wages and prices fail to clear markets, the third wave of new Keynesian research aimed to explain why this was the case. Various hypotheses were explored: that firms face "menu costs" when they choose to change their prices; that firms pay their workers "efficiency wages" above the market-clearing level to increase worker productivity; and that wage and price setters deviate from perfect rationality. Mankiw (1985) and Akerlof and Yellen (1985) pointed out that when firms have market power, there are large differences between the private and social cost–benefit calculations regarding price adjustment, so a sticky-price equilibrium could be privately rational (or near rational) while socially very costly. Blanchard and Kiyotaki (1987) showed that part of this divergence between private and social incentives results from an aggregate-demand externality: when one firm cuts its prices, it increases real money balances and thus the demand for the products of all firms. Ball and Romer (1990) established that there is strong complementary between real and nominal rigidities, so any motive for avoiding relative-price changes would exacerbate the sluggishness of nominal prices.

In retrospect, these various new Keynesian contributions were more related and complementary than they seemed at the time, even to people working on them. For example, it is tempting to see the early work on general disequilibrium as a dead end—a research program that sowed the seeds of its own demise by its assumption of predetermined prices. Indeed, this work rarely finds its way on to reading lists today. Yet one can also see a progression of related ideas about how the economy works when prices do not move instantly to balance supply and demand.

There is, for instance, an interesting but rarely noticed relationship between the first and third waves of new Keynesian economics. In particular, one can view the third wave as establishing the centrality of the Keynesian regime highlighted in the first wave. When firms have market power, they charge prices above marginal cost, so they always want to sell more at prevailing prices. In a sense, if all firms have some degree of market power, then goods markets are typically in a state of excess supply. This theory of the goods market is often married to a theory of the labor market with above-equilibrium wages, such as the efficiency-wage model. In this case, the "Keynesian" regime of generalized excess supply is not just one possible outcome for the economy, but the typical one.

In my judgment, these three waves of new Keynesian research added up to a coherent microeconomic theory for the failure of the invisible hand to work for short-run macroeconomic phenomena. We understand how markets interact when there are price rigidities, the role that expectations can play, and the incentives that price setters face as they choose whether or not to change prices. As a matter of science, there was much success in this research (although, as a participant, I cannot claim to be entirely objective). The work was not revolutionary, but it was not trying to be. Instead, it was counterrevolutionary: its aim was to defend the essence of the neoclassical–Keynesian synthesis from the new classical assault.

Was this work also successful as a matter of engineering? Did it help policy-makers devise better policies to cope with the business cycle? The judgment here must be less positive—a topic to which I will return shortly.

But it is remarkable that the new Keynesians were, by temperament, more inclined to become macroeconomic engineers than were economists working within the new classical tradition. Among the leaders of the new classical school, none (as far as I know) has ever left academia to take a significant job in public policy. By contrast, the new Keynesian movement, like the earlier generation of Keynesians, was filled with people who traded a few years in the ivory tower for a stay in the nation's capital. Examples include Stanley Fischer, Larry Summers, Joseph Stiglitz, Janet Yellen, John Taylor, Richard Clarida, Ben Bernanke, and myself. The first four of these economists came to Washington during the Clinton years; the last four during the Bush years. The division of economists between new classicals and new Keynesians is not, fundamentally, between the political right and the political left. To a greater extent, it is a split between pure scientists and economic engineers.

Digression and Vitriol

The theory and empirics of long-run economic growth are beyond the scope of the essay, but it is worth pointing out that these topics occupied much of the attention of macroeconomists during the decade of the 1990s. This work drew attention away from short-run fluctuations, which had dominated the field of macroeconomics since its birth half a century earlier.

There are several reasons for the emergence of growth as a major area for research. First, a series of influential papers by Paul Romer (1986) and others offered a new set of ideas and tools for analyzing what is surely one of the most compelling topics in economics—the large gap between rich and poor nations. Second, new cross-country data became available that allowed systematic examination of the validity of alternative theories (Summers and Heston, 1991). Third, the U.S. economy in the 1990s was experiencing its longest expansion in history. Just as the early Keynesians were attracted to the field because of its immediate relevance to the nation's health, the economy of the 1990s suggested to that generation of students that the business cycle was no longer of great practical importance.

There is also a fourth, more troublesome reason why budding macroeconomists of the 1990s were drawn to study long-run growth rather than short-run fluctuations: the tension between new classical and new Keynesian worldviews. While Lucas, the leading new classical economist, was proclaiming that "people don't take Keynesian theorizing seriously anymore," leading Keynesians were equally patronizing to their new classical colleagues. In his Presidential Address to the American Economic Association, Solow (1980) called it "foolishly restrictive" for the new classical economists to rule out by assumption the existence of wage and price rigidities and the possibility that markets do not clear. He said, "I remember reading once that it is still not understood how the giraffe manages to pump an adequate blood supply all the way up to its head; but it is hard to imagine that anyone would therefore conclude that giraffes do not have long necks."

In an interview with Arjo Klamer (1984) a few years later, Lucas remarked, "I don't think that Solow, in particular, has ever tried to come to grips with any of these issues except by making jokes." In his own interview in the same volume, Solow explained his unwillingness to engage with the new classical economists: "Suppose someone sits down where you are sitting right now and announces to me that he is Napoleon Bonaparte. The last thing I want to do with him is to get involved in a technical discussion of cavalry tactics at the Battle of Austerlitz. If I do that, I'm getting tacitly drawn into the game that he is Napoleon Bonaparte."

To some extent, this dispute reflects the differing perspectives of the protagonists about the goal of the field. Lucas seems to be complaining that Solow does not appreciate the greater analytic rigor that new classical macroeconomics can offer. Solow seems to be complaining the Lucas does not appreciate the patent lack of reality of his market-clearing assumptions. They each have a point. From the standpoint of science, the greater rigor that the new classicals offered has much appeal. But from the standpoint of engineering, the cost of this added rigor seems too much to bear.

I dwell on the nature of this debate not only because it reflects the underlying tension between scientists and engineers but also because it helps explain the choices made by the next generation of economists. Such vitriol among intellectual giants attracts attention, much in the way that the patrons in a bar gather around a fistfight, egging on the participants. But it was not healthy for the field of macroeconomics. Not surprisingly, many young economists chose to avoid taking sides in this dispute by turning their attention away from economic fluctuations and toward other topics.

A New Synthesis, or a Truce?

An old adage holds that science progresses funeral by funeral. Today, with the benefits of longer life expectancy, it would be more accurate (if less vivid) to say that science progresses retirement by retirement. In macroeconomics, as the older generation of protagonists has retired or neared retirement, it has been replaced by a younger generation of macroeconomists who have adopted a culture of greater civility. At the same time, a new consensus has emerged about the best way to understand economic fluctuations. Marvin Goodfriend and Robert King (1997) have dubbed this consensus view "the new neoclassical synthesis." This synthesis model has been widely applied in research on monetary policy (Clarida, Gali, and

Gertler, 1999; McCallum and Nelson, 1999). The most extensive treatment of this new synthesis is Michael Woodford's (2003) monumental (in both senses of the word) treatise.

Like the neoclassical–Keynesian synthesis of an earlier generation, the new synthesis attempts to merge the strengths of the competing approaches that preceded it. From the new classical models, it takes the tools of dynamic stochastic general equilibrium theory. Preferences, constraints, and optimization are the starting point, and the analysis builds up from these microeconomic foundations. From the new Keynesian models, it takes nominal rigidities and uses them to explain why monetary policy has real effects in the short run. The most common approach is to assume monopolistically competitive firms that change prices only intermittently, resulting in price dynamics sometimes called the new Keynesian Phillips curve. The heart of the synthesis is the view that the economy is a dynamic general equilibrium system that deviates from a Pareto optimum because of sticky prices (and perhaps a variety of other market imperfections).

It is tempting to describe the emergence of this consensus as great progress. In some ways, it is. But there is also a less sanguine way to view the current state of play. Perhaps what has occurred is not so much a synthesis as a truce between intellectual combatants, followed by a face-saving retreat on both sides. Both new classicals and new Keynesians can look to this new synthesis and claim a degree of victory, while ignoring the more profound defeat that lies beneath the surface.

The heart of this new synthesis—a dynamic general equilibrium system with nominal rigidities—is precisely what one finds in the early Keynesian models. Hicks proposed the IS–LM model, for example, in an attempt at putting the ideas of Keynes into a general equilibrium setting. (Recall that Hicks won the 1972 Nobel Prize jointly with Kenneth Arrow for contributions to general equilibrium theory.) Klein, Modigliani, and the other model builders were attempting to bring that general equilibrium system to the data to devise better policy. To a large extent, the new synthesis picks up the research agenda that the profession abandoned, at the behest of the new classicals, in the 1970s.

With the benefit of hindsight, it is clear that the new classical economists promised more than they could deliver. Their stated aim was to discard Keynesian theorizing and replace it with market-clearing models that could be convincingly brought to the data and then used for policy analysis. By that standard, the movement failed. Instead, they helped to develop analytic tools that are now being used to develop another generation of models that assume sticky prices and that, in many ways, resemble the models that the new classicals were campaigning against.

The new Keynesians can claim a degree of vindication here. The new synthesis discards the market-clearing assumption that Solow called "foolishly restrictive" and that the new Keynesian research on sticky prices aimed to undermine. Yet the new Keynesians can be criticized for having taken the new classicals' bait and, as a result, pursuing a research program that turned out to be too abstract and insufficiently practical. Paul Krugman (2000) offers this evaluation of the new Keynesian research program: "One can now explain how price stickiness *could* happen. But useful predictions about when it happens and when it does not, or models that build from menu costs to a realistic Phillips curve, just don't seem to be forthcoming." Even as a proponent of this line of work, I have to admit that there is some truth to that assessment.

The View from Central Banking

If God put macroeconomists on earth to solve practical problems, then Saint Peter will ultimately judge us by our contributions to economic engineering. So let's ask: Have the developments in business cycle theory over the past several decades improved the making of economic policy? Or, to set a more modest goal, have the advances in macroeconomic science altered how economic policy is analyzed and discussed among professional economists who are involved in the policy process?

One place to find evidence to answer these questions is Laurence Meyer's charming memoir *A Term at the Fed*. In 1996, Meyer left his job as an economics professor at Washington University and as a prominent economic consultant to serve for six years as a governor of the Federal Reserve. His book provides a window into how economists at the highest reaches of monetary policymaking view their jobs and the approaches they take to analyzing the economy.

The book leaves the reader with one clear impression: recent developments in business cycle theory, promulgated by both new classicals and new Keynesians, have had close to zero impact on practical policymaking. Meyer's analysis of economic fluctuations and monetary policy is intelligent and nuanced, but it shows no traces of modern macroeconomic theory. It would seem almost completely familiar to someone who was schooled in the neoclassical–Keynesian synthesis that prevailed around 1970 and has ignored the scholarly literature ever since. Meyer's worldview would be easy to dismiss as outdated if it were idiosyncratic, but it's not. It is typical of economists who have held top positions in the world's central banks.

It is fashionable among academics to believe that central banking has been strongly influenced by the rules-versus-discretion literature, particularly the work on time inconsistency that started with Kydland and Prescott (1977). Two institutional changes are often linked with these academic contributions: the increased independence of central banks in countries such as New Zealand and the adoption of inflation targeting as a policy regime in many central banks around the world. These institutional changes, in turn, are then linked to improvements in monetary policy. According to this line of argument, Kydland and Prescott are to be thanked for the low, stable inflation that many countries have enjoyed over the past two decades.

This self-congratulatory view runs into two problems. First, the institutional changes we have observed are at best loosely connected to the issues raised in the theoretical literature. An independent central bank is not the same as a rule-bound central bank. The U.S. Federal Reserve has long had a high degree of independence without ever committing itself to a policy rule. Even inflation targeting is closer to a statement of intentions and a way of communicating with the public than it is a commitment to a policy rule. Bernanke (2003) has called it "constrained discretion."

The second, more significant problem is that these institutional changes are not necessarily linked to the improvements we have witnessed in monetary policy. Ball and Sheridan (2005) look at a large sample of countries and show that adoption of inflation targeting does not help explain the recent move toward low, stable inflation. Monetary policy has improved both in those counties that have adopted inflation targets and in those that have not. This worldwide improvement in inflation outcomes could be because the world economy has not had to deal with supply shocks as adverse as those experienced in the 1970s or because central bankers have learned from the experience of the 1970s that high inflation should be assiduously avoided. But the evidence shows that inflation targeting is not a prerequisite for good monetary policy.

The Federal Reserve under Alan Greenspan is a case in point. According to Blinder and Reis (2005), Greenspan has a rightful claim to be "the greatest central banker who ever lived." Indeed, by most accounts, monetary policy worked remarkably well under his leadership. Yet throughout his time at the helm of the Fed, Greenspan avoided any announcement of a policy rule, valuing flexibility over commitment. Here is how Greenspan (2003) defended his choice: "Some critics have argued that such an approach to policy is too undisciplined—judgmental, seemingly discretionary, and difficult to explain. The Federal Reserve should, some conclude, attempt to be more formal in its operations by tying its actions solely to the prescriptions of a formal policy rule. That any approach along these lines would lead to an improvement in economic performance, however, is highly doubtful . . . . Rules by their nature are simple, and when significant and shifting uncertainties exist in the economic environment, they cannot substitute for risk-management paradigms, which are far better suited to policymaking." Yet, despite Greenspan's aversion to policy rules, inflation was low and stable during his tenure as Fed chairman. Greenspan proves that central banks can produce desirable outcomes while wielding substantial discretionary powers.

The View from Fiscal Policy

Another place to look for the practical impact of macroeconomic theory is the analysis of fiscal policy. The Bush tax cuts of 2001 and 2003 offer a good case study, in part because they are a recent attempt at a major fiscal stimulus to combat a recession and in part because, as chairman of the Council of Economic Advisers for two years, I am familiar with much of the economic analysis that laid the foundation for this policy. To be sure, there were many motives for the design of the Bush tax policy. The expansion of the child tax credit, for example, was rooted as much in politics and social philosophy as it was in economics. But economists at the Council of Economic Advisers and Treasury had substantial input into the development of the policy, so it is illuminating to consider the tools they brought to the job.

The economic analysis of the Bush tax plan was done with one eye on long-run growth and one eye on the short-run business cycle. The long-run perspective would be familiar to students of public finance. Most significantly, in 2003 Bush proposed eliminating the double taxation of income from corporate capital. The final bill passed by Congress did not fully achieve this goal, but the substantial cut in tax rates on dividends moved in the direction of greater tax neutrality, reducing the bias for retained earnings over dividends, the bias for debt over equity finance, and the bias for noncorporate over corporate capital. It also moved the tax code further in the direction of taxing consumption rather than income. This latter goal is consistent with a well-established literature in public finance (for example, Diamond and Mirrlees, 1971; Atkinson and Stiglitz, 1976; Feldstein, 1978; Chamley, 1986) and is not particularly new as a matter of economic theory. Three decades ago, Atkinson and Stiglitz noted, even then, there was a "conventional presumption in favor of consumption rather than income taxation."

More relevant to this essay, however, is the short-run analysis of tax policy. As President George W. Bush took office in 2001, the economy was heading into a recession after the bursting of the stock market bubble of the late 1990s. One goal of the tax cuts was to stimulate economic recovery and employment. When Bush signed the Jobs and Growth Tax Relief Reconciliation Act of 2003, he explained the policy as follows: "When people have more money, they can spend it on goods and services. And in our society, when they demand an additional good or a service, somebody will produce the good or a service. And when somebody produces that good or a service, it means somebody is more likely to be able to find a job." This logic is quintessentially Keynesian.

The Council of Economic Advisers was asked to quantify how tax relief would affect employment. We answered this question using a mainstream macroeconometric model. The specific model we used while I was there was the one maintained by Macroeconomic Advisers, the consulting firm created and run by Laurence Meyer before he was a Federal Reserve governor. This model was being used by the staff of the Council of Economic Advisers long before I arrived as chairman and, in fact, had been used for almost two decades under both Republican and Democratic administrations. The choice of this particular model is not crucial, however, for the Macroeconomic Advisers model is similar to other large macroeconometric models, such as the FRB/US model maintained by the Federal Reserve. From the standpoint of intellectual history, these models are the direct descendents of the early modeling efforts of Klein, Modigliani and Eckstein. Research by new classicals and new Keynesians has had minimal influence on the construction of these models.

The real world of macroeconomic policymaking can be disheartening for those of us who have spent most of our careers in academia. The sad truth is that the macroeconomic research of the past three decades has had only minor impact on the practical analysis of monetary or fiscal policy. The explanation is not that economists in the policy arena are ignorant of recent developments. Quite the contrary: the staff of the Federal Reserve includes some of the best young Ph.D.s, and the Council of Economic Advisers under both Democratic and Republican administrations draws talent from the nation's top research universities. The fact that modern macroeconomic research is not widely used in practical policymaking is prima facie evidence that it is of little use for this purpose. The research may have been successful as a matter of science, but it has not contributed significantly to macroeconomic engineering.

Inside the Classroom

Beyond the corridors of power in the world's capitals, there is another place where the economics profession tries to sell its wares to a broader audience—the undergraduate classroom. Those of us who regularly teach undergraduates see our job as producing citizens who are well-informed about the principles of good policy. Our choice of material is guided by what we see as important for the next generation of voters to understand.

Like policymakers, undergraduates typically have little interest in theory for theory's sake. Instead, they are interested in understanding how the real world works and how public policy can improve economic performance. Except for the rare student who is considering graduate school and a career as an academic economist, the undergraduate has the perspective of an engineer, more than that of a scientist. It is, therefore, useful to take note of what we choose to teach undergraduates. And there is no better place to see what we teach than in the contents of the most widely used undergraduate textbooks.

Consider, for example, the books used to teach intermediate-level macroeconomics. A generation ago, the three leading texts for this course were those by Robert Gordon; Robert Hall and John Taylor; and Rudiger Dornbusch and Stanley Fischer. Today, the top three sellers are those written by Olivier Blanchard; Andrew Abel and Ben Bernanke; and myself. The common thread is that each of these six books was written by at least one economist with graduate training from MIT, a prominent engineering school where the dominant macroeconomic tradition was that of Samuelson and Solow. In all these books, the basic theory taught to undergraduates is some version of aggregate demand and aggregate supply, and the basic theory of aggregate demand is the IS–LM model. The same lesson can be gleaned by perusing the most widely used textbooks for freshman-level economics: short-run economic fluctuations are best understood using some version of the neoclassical–Keynesian synthesis.

I do not mean to suggest that pedagogy has been stagnant as the field has evolved. Today's textbooks place greater emphasis on classical monetary theory, models of long-run growth, and the role of expectations than did those of 30 years ago. There is less confidence about what policy can accomplish and more emphasis on policy rules over discretionary monetary and fiscal actions (despite the lack of evidence on the practical importance of policy rules). But the basic framework that modern students learn to make sense of the business cycle is one that would be familiar to an early generation of Keynesians.

The exception that proves the rule is the intermediate text written by Robert Barro, first published in 1984. Barro's book provided a clear and accessible introduction to the new classical approach to macroeconomics aimed at undergraduates. Keynesian models were included, but they were covered late in the book, briefly, and with little emphasis. When the book came out, it received substantial attention and acclaim. However, while many macroeconomists read the Barro book and were impressed by it, many fewer chose it for their students. The new classical revolution in pedagogy that Barro hoped to inspire never took off, and the Barro text did not offer significant competition to the dominant textbooks of the time.

This lack of revolution in macroeconomic pedagogy stands in stark contrast to what occurred half a century ago. When Samuelson's introductory textbook was first published in 1948 with the aim of introducing undergraduates to the Keynesian revolution, the world's teachers rapidly and heartily embraced the new approach. By contrast, the ideas of new classicals and new Keynesians have not fundamentally changed how undergraduate macroeconomics is taught.

Not a Dentist in Sight

John Maynard Keynes (1931) famously opined, "If economists could manage to get themselves thought of as humble, competent people on a level with dentists, that would be splendid." He was expressing a hope that the science of macroeconomics would evolve into a useful and routine type of engineering. In this future utopia, avoiding a recession would be as straightforward as filling a cavity.

The leading developments in academic macroeconomics of the past several decades bear little resemblance to dentistry. New classical and new Keynesian research has had little impact on practical macroeconomists who are charged with the messy task of conducting actual monetary and fiscal policy. It has also had little impact on what teachers tell future voters about macroeconomic policy when they enter the undergraduate classroom. From the standpoint of macroeconomic engineering, the work of the past several decades looks like an unfortunate wrong turn.

Yet from the more abstract perspective of macroeconomic science, this work can be viewed more positively. New classical economists were successful at showing the limitations of the large Keynesian macroeconometric models and the policy prescriptions based on these models. They drew attention to the importance of expectations and the case for policy rules. New Keynesian economists have supplied better models to explain why wages and prices fail to clear markets and, more generally, what types of market imperfections are needed to make sense of short-run economic fluctuations. The tension between these two visions, while not always civil, may have been productive, for competition is as important to intellectual advance as it is to market outcomes.

The resulting insights are being incorporated into the new synthesis that is now developing and which will, eventually, become the foundation for the next generation of macroeconometric models. For those of us interested in macroeconomics as both science and engineering, we can take the recent emergence of a new synthesis as a hopeful sign that more progress can be made on both fronts. As we look ahead, "humble" and "competent" remain ideals toward which macroeconomists can aspire.

■ *I am grateful to Steven Braun, James Hines, Donald Marron, David Romer, Andrei Shleifer, Timothy Taylor, Michael Waldman, and Noam Yuchtman for helpful comments.*

References

Akerlof, George A. and Janet L. Yellen. 1985. "A Near-Rational Model of the Business Cycle with Wage and Price Inertia." *Quarterly Journal of Economics.* 100:5 (Supplement), pp. 823–38.

Atkinson, A. B. and J. E. Stigltiz. 1976. "The Design of Tax Structure: Direct versus Indirect Taxation," *Journal of Public Economics.* 6:1–2, pp. 55–75.

Ball, Laurence and David Romer. 1990. "Real Rigidities and the Non-Neutrality of Money." *Review of Economic Studies.* 57:2, pp. 183–203.

Ball, Laurence and Niamh Sheridan. 2005. "Does Inflation Targeting Matter?" in *The Inflation-Targeting Debate.* Ben S. Bernanke and Michael Woodford, eds. Chicago: University of Chicago Press, pp. 249–82.

Barro, Robert. 1977. "Unanticipated Money Growth and Unemployment in the United States." *American Economic Review.* March, 67:2, pp. 101–15.

Barro, Robert J. and Herschel Grossman. 1971. "A General Disequilibrium Model of Income and Employment." *American Economic Review.* March, 61:1, pp. 82–93.

Bernanke, Ben S. 2003. "Constrained Discretion and Monetary Policy." Remarks before the Money Marketeers of New York University, New York, New York. February 3.

Blanchard, Olivier Jean, and Nobuhiro Kiyotaki. 1987. "Monopolistic Competition and the Effects of Aggregate Demand." *American Economic Review.* September, 77:4, pp. 647–66.

Blinder, Alan S. and Ricardo Reis. 2005. "Understanding the Greenspan Standard,“ Paper presented at Federal Reserve Bank of Kansas City conference at Jackson Hole. (Princeton University.)

Breit, William and Barry T. Hirsch. 2004. *Lives of the Laureates, 4th edition.* Cambridge, MA: MIT press.

Chamley, Christophe. 1986. "Optimal Taxation of Capital Income in General Equilibrium with Infinite Lives." *Econometrica.* May, 54:3, pp. 607–22.

Clarida, Richard, Jordi Gali, and Mark Gertler. 1999. "The Science of Monetary Policy: A New Keynesian Perspective." *Journal of Economic Literature.* December, 37:4, pp. 1661–1707.

Diamond, Peter A. and James A. Mirrlees. 1971. "Optimal Taxation and Public Production II: Tax Rules. *American Economic Review.* June, 61:3, pp. 261–78.

Feldstein, Martin. 1978. "The Welfare Cost of Capital Income Taxation," *Journal of Political Economy.* April, 86:2, pt. 2, pp. S29–S51.

Fischer, Stanley. 1977. "Long-term Contracts, Rational Expectations, and the Optimal Money Supply Rule." *Journal of Political Economy.* February, 85:1, pp. 191–205.

Friedman, Milton. 1957. *A Theory of the Consumption Function.* Princeton: Princeton University Press.

Friedman, Milton. 1968. "The Role of Monetary Policy." *American Economic Review.* March, 58:1, pp. 1–17.

Freidman, Milton and Anna Jacobson Schwartz. 1963. *A Monetary History of the United States, 1867–1960.* Princeton: Princeton University Press.

Garfield, Eugene. 1990. "Who Will Win the Nobel Prize in Economics? Here's a Forecast Based on Citation Indicators," in *Essays of an Information Scientist: Journalology, KeyWords Plus, and Other Essays, Vol. 13.* Philadelphia, PA: ISS Press, p. 83–87.

Goodfriend, Marvin and Robert King. 1997. "The New Neoclassical Synthesis and the Role of

Monetary Policy," in *NBER Macroeconomics Annual.* Ben S. Bernake and Julio J. Rotemberg, eds. Cambridge, MA: MIT Press. pp. 231–83.

Greenspan, Alan. 2003. "Monetary Policy under Uncertainty." Remarks at a symposium sponsored by the Federal Reserve Bank of Kansas City, Jackson Hole, Wyoming, August 29.

Hicks, John R. 1937. "Mr. Keynes and the 'Classics.' " *Econometrica.* April, 5:2, pp. 147–59.

Hume, David. 1752. "Of Money," in *Essays.* London: George Routledge and Sons.

Keynes, John Maynard. 1931. *Essays in Persuasion.* New York: Norton.

Klamer, Arjo. 1984. *Conversations with Economists.* Totowa, NJ: Rowman and Allanheld.

Klein, Lawrence R. 1946. "Macroeconomics and the Theory of Rational Behavior." *Econometrica.* April, 14:2, pp. 93–108.

Krugman, Paul. 2000. "How Complicated Does the Model Have to Be?" *Oxford Review of Economic Policy.* 16:4, pp. 33–42.

Kydland, Finn and Edward C. Prescott. 1977. "Rules Rather Than Discretion: The Inconsistency of Optimal Plans." *Journal of Political Economy.* June, 85:3, pp. 473–92.

Kydland, Finn and Edward C. Prescott. 1982. "Time to Build and Aggregate Fluctuations." *Econometrica.* November, 50:6, pp. 1345–71.

Long, John. B. and Charles Plosser. 1983. "Real Business Cycles." *Journal of Political Economy.* February, 91:1, pp. 39–69.

Lucas, Robert E, Jr. 1973. "Some International Evidence on Output–Inflation Tradeoffs." *American Economic Review.* June, 63:3, pp. 326–34.

Lucas, Robert E. Jr. 1976. "Econometric Policy Evaluation: A Critique." *Carnegie-Rochester Conference Series on Public Policy.* 1, pp. 19–46.

Lucas, Robert E., Jr. 1980. "The Death of Keynesian Economics." *Issues and Ideas.* (University of Chicago, Chicago, IL). Winter, pp. 18–19.

Lucas, Robert E. Jr. and Thomas J. Sargent. 1979. "After Keynesian Macroeconomics." *Federal Reserve Bank of Minneapolis Quarterly Review.* Spring, 3:2, pp. 1–16.

Malinvaud, Edmund. 1977. *The Theory of Unemployment Reconsidered.* Oxford, UK: Blackwell.

Mankiw, N. Gregory. 1985. "Small Menu Costs and Large Business Cycles: A Macroeconomic Model of Monopoly." *Quarterly Journal of Economics.* May, 100:2, pp. 529–37.

McCallum, Bennett T. and Edward Nelson. 1999. "An Optimizing IS–LM Specification for Monetary Policy and Business Cycle Analysis." *Journal of Money, Credit, and Banking.* August, 31:3, pt. 1, pp. 296–316.

Meyer, Laurence H. 2004. *A Term at the Fed: An Insider's View.* New York: Harper-Collins.

Modigliani, Franco. 1944. "Liquidity Preference and the Theory of Interest and Money." *Econometrica.* January, 12:1, pp. 45–88.

Phelps, Edmund. 1968. "Money Wage Dynamics and Labor Market Equilibrium." *Journal of Political Economy.* July/August, 76:4, pt. 2, pp. 678–711.

Pigou, Arthur. 1927. *Industrial Fluctuations.* London: Macmillan and Company.

Quandt, Richard E. 1976. "Some Quantitative Aspects of the Economics Journal Literature." *Journal of Political Economy.* August, 84:4, pt. 1, pp. 741–55.

Romer, Paul. 1986. "Increasing Returns and Long Run Growth." *Journal of Political Economy.* October, 94:5, pp. 1002–37.

Samuelson, Paul A. 1948. *Economics: An Introductory Analysis.* New York: McGraw-Hill.

Samuelson, Paul. A. 1988. "Keynesian Economics and Harvard: In the Beginning." *Challenge: The Magazine of Economic Affairs.* July/August, 31, pp. 32–34.

Samuelson, Paul A. and Robert M. Solow. 1960. "Analytical Aspects of Anti-Inflation Policy." *American Economic Review.* May, 50:2, pp. 177–194.

Sargent, Thomas J. and Neil Wallace. 1975. "Rational Expectations, the Optimal Monetary Instrument, and the Optimal Money Supply Rule." *Journal of Political Economy.* April, 83:2, pp. 241–54

Solow, Robert M. 1980. "On Theories of Unemployment." *American Economic Review.* March, 70:1, pp. 1–11.

Summers, Robert and Alan Heston. 1991. "The Penn World Table (Mark 5): An Expanded Set of International Comparisons, 1950–1988." *Quarterly Journal of Economics.* May, 106:2, pp. 327–68.

Taylor, John B. 1980. "Aggregate Dynamics and Staggered Contracts." *Journal of Political Economy.* February, 88:1, pp. 1–23.

Woodford, Michael. 2003. *Interest and Prices.* Princeton: Princeton University Press.

MATLAB TUTORIAL

The idea behind this tutorial is that you view it in one window or have a printout of it in front of you while running Matlab in another window. You should be able to re-do all of the plots and calculations in the tutorial by copying the text from the tutorial into Matlab or an m-file. `I will use this font to denote text that has to be typed in Matlab or the text that Matlab returns.`

Index

1. Getting Started

To start Matlab from Windows:

1) Click on Start
2) Select Programs and then Matlab
3) Click on the Matlab icon and this will bring up the window called the Command Window. This is the main space you will be using, and it will look as follows:

```
To get started, type one of these: helpwin, helpdesk, or demo. For
product information, type tour or visit www.mathworks.com.

»
```

This window allows a user to enter simple commands. To perform a simple computations type a command and next press the **Enter** or **Return** key. For instance

```
» s = 1 + 2

s =

     3
```

Note that the results of these computations are saved in variables whose names are chosen by the user. If they will be needed during your current MATLAB session, you can obtain their values typing their names and pressing the **Enter** or **Return** key. For instance, if you type again:

```
» s

s =

    3
```

Some useful format commands for the display are:

- `format short` 3.1416
- `format long` 3.14159265358979
- `format compact` suppresses extra line feeds

To close MATLAB type exit in the **Command Window** and next press **Enter** or **Return** key. A second way to close your current MATLAB session is to select **File** in the MATLAB's toolbar and next click on **Exit MATLAB** option. All unsaved information residing in the MATLAB **Workspace** will be lost. You can also exit by typing:

```
» quit
```

2. Matlab Help

One of the nice features of MATLAB is its help system.. A good place to start is with the command:

» `help help` that explains how the help systems works. In addition, the command

» `help` produces a list of topics for which help is available. The list is printed at the end of the tutorial. Looking at it we find, for example, the entry:

```
matlab\elfun          -  Elementary math functions.
```

» `help elfun` produces then a list of the math functions available. Alternatively, the `helpwin` command opens a new window on the screen. To find an information you need double click on the name of the subdirectory and next double click on a function to see the help text for that function. In general, to get more information on any given command, type:

```
» help command
```

3. Numbers, Arithmetic Operations and Special Characters

There are three kinds of numbers used in MATLAB:

- integers
- real numbers
- complex numbers

In addition to these, MATLAB has three variables representing non-numbers:

```
-Inf  Inf   NaN
```

The **–Inf** and **Inf** are the negative and positive infinity respectively. Infinity is generated by overflow or by the operation of dividing by zero. The **NaN** stands for Not-A-Number and it is obtained as a result of the mathematically undefined operations such as 0.0/0.0.

The list of basic arithmetic operations in MATLAB includes six operations:

+ : addition

- : substraction

* : multiplication

/ : right division

\ : left division

^ : power

Further, there is a menu of special characters that have specific uses. The main ones are:

- Equal ==

- Not equal ~=

- Less than <

- Greater than >

- Less than or equal <=

- Greater than or equal >=

- Logical AND &

- Logical OR |

- Logical NOT ~

- Colon :

- Parentheses and subscripting ( )

- Brackets [ ]

- Decimal point .

- Continuation ...

- Separator ,

- Semicolon ; (suppresses the output of a calculation)

- Comment %

- Assignment =

- Quote ' *statement* '

- Transpose '

4. Vectors, Matrices and Three Dimensional Arrays

All inputs in in MATLAB are taken to be arrays. In this way, a scalar is a *1x1* array, a row vector is a *1xn* array, a column vector is an *nx1* array and a matrix is an *nxm* array. MATLAB can also handle three dimensional arrays (that you can think of as matrices stacked one on top of the other). To define arrays, we always use the brackets.

Vectors

To create the row vector a type:

```
» a = [1 2 3 4 5 6 9 8 7]

a =

     1     2     3     4     5     6     9     8     7
```

Note that separating the elements by a space you create a row vector. If you want to create a column vector, separate the elements by semicolons:

```
» b = [1; 2; 3; 4; 5; 6]

b =

     1
     2
     3
     4
     5
     6
```

Another way to create column vectors is by using the transpose, which is denoted with a ' after the bracket:

```
» b = [1  2   3  4  5  6]'
```

The command length returns the number of components of a vector:

```
» length(a)


ans =

     9
```

If you ant to type a scalar just write:

```
» c = 2

c =
     2
```

Suppose you would like to add 2 to each of the elements in vector a. The equation for that looks like:

```
» e = a + 2

e =

           3      4      5      6      7      8     11     10      9
```

Now suppose, you would like to add two vectors together. If the two vectors are the same length, it is easy. Simply add the two:

```
» f = a + e

f =

     4      6      8     10     12     14     20     18     16
```

Subtraction of vectors of the same length works exactly the same way. A useful command is ``whos", which displays the names of all defined variables and their types:

```
>> whos

Name       Size          Bytes  Class

  a          1x9              72  double array
  c          1x1               8  double array
  e          1x9              72  double array
  f          1x9              72  double array
  t          1x11             88  double array

Grand total is 39 elements using 312 bytes
```

Note that each of these variables is an vector. The "shape" of the vector determines its exact type. The scalar c is a 1 x 1 array, the vector a is a 1 x 9 array.

Matrices

Entering matrices into Matlab is the same as entering a vector, except each row of elements is separated by a semicolon **(;)** or a return:

```
» B = [1 2 3 4; 5 6 7 8; 9 10 11 12]

     B =
          1    2    3     4
          5    6    7     8
          9   10   11    12
```

Alternatively, you can enter matrices as follows:

```
» B = [ 1   2   3   4
        5   6   7   8
        9 10  11  12]

    B =
          1    2    3     4
          5    6    7     8
          9   10   11    12
```

Like with vectors, we have an in-built function that gives the dimensions of a matrix:

```
 » size(B)

ans =

     3   4
```

Creating special matrices

- zeros(m,n) creates an **m x n** matrix of zeros
- ones(m,n) creates an **m x n** matrix of ones
- eye(n) creates the **n x n** identity matrix
- diag(v) (assuming v is an n-vector) creates an **n x n** diagonal matrix with v on the diagonal.

```
» zeros(2,1)

ans =

     0
     0
```

Matrix manipulation and operations

Matrices in Matlab can be manipulated in many ways. For one, you can find the transpose of a matrix using the apostrophe key:

```
» C = B'

      C =
               1    5     9
               2    6    10
               3    7    11
               4    8    12
```

Now you can multiply the two matrices B and C together. Remember that order matters when multiplying matrices.

```
» D = B * C
      D =       30     70    110
                70    174    278
               110    278    446

» D  = C * B

       D =
           107   122   137   152
           122   140   158   176
           137   158   179   200
           152   176   200   224
```

Another option for matrix manipulation is that you can multiply the corresponding elements of two matrices using the .* operator (the matrices must be the same size to do this).

```
» E = [1 2;3 4]

       E =

            1     2
            3     4

» F = [2 3;4 5]

       F =

            2     3
            4     5

» G = E .* F

      G =

           2     6
          12    20
```

If you have a square matrix, like E, you can also multiply it by itself as many times as you like by raising it to a given power.

```
» E^3
      ans =
            37    54
            81   118
```

If wanted to cube each element in the matrix, just use the element-by-element cubing.

```
» E.^3
           ans =
                1      8
               27     64
```

In general:

```
a .* b
```
multiplies each element of a by the respective element of b
```
a ./ b
```
divides each element of a by the respective element of b
```
a .\ b
```
divides each element of b by the respective element of a
```
a .^ b
```
raise each element of a by the respective b element

You can also find the inverse of a matrix:

```
» X = inv(E)

           X =
              -2.0000    1.0000
               1.5000  -0.5000
```

or its eigenvalues and eigenvectors:

```
» eig(E)

      ans =

         -0.3723
          5.3723

» [vec,va]=eig(E)

      vec =

         -0.8246   -0.4160
          0.5658   -0.9094


      va =

         -0.3723         0
               0    5.3723
```

If A and B are matrices, then Matlab can compute A+B and A-B *when these operations are defined*. For example, consider the following commands:

```
>> A = [1 2 3;4 5 6;7 8 9]
A =

     1     2     3
     4     5     6
     7     8     9
```

```
>> B = [1 1 1;2 2 2;3 3 3]
B =

     1     1     1
     2     2     2
     3     3     3

>> C = [1 2;3 4;5 6]
C =

     1     2
     3     4
     5     6

>> A+B

ans =
     2     3     4
     6     7     8
    10    11    12

>> A+C

??? Error using ==> +
Matrix dimensions must agree.
```

The Colon Operator

The colon operator **:** can be used to perform special and useful operations. In particular, you will be able to use it to extract or manipulate elements of matrices. The following command creates a row vector whose components increase arithmetically:

```
>> 1:5

ans =
     1     2     3     4     5
```

And, if three numbers, integer or non-integer, are separated by two colons, the middle number is interpreted to be a "range" and the first and third are interpreted to be "limits". Thus

```
» b = 0.0 : 0.2 : 1.0

b =
         0    0.2000    0.4000    0.6000    0.8000    1.0000
```

Suppose you want to create a vector with elements between 0 and 20 evenly spaced in increments of 2. Then you have to type:

```
» t = 0:2:20

t =

     0     2     4     6     8    10    12    14    16    18    20
```

A negative step is also allowed. The command linspace has similar results; it creates a vector with linearly spaced entries.

```
» help linspace

 LINSPACE Linearly spaced vector.
    LINSPACE(x1, x2) generates a row vector of 100 linearly
    equally spaced points between x1 and x2.

 LINSPACE(x1, x2, N) generates N points between x1 and x2.
```

The colon operator can also be used to create a vector from a matrix. Define:

```
» x = [ 2  6  8
        0  1  7
       -2  5 -6 ]

» y = x(:,1)

y =
     2
     0
    -2
```

Note that the expressions before the comma refer to the matrix rows and after the comma to the matrix columns. The same applies to 3D arrays, where the third index indicates the third dimension.

```
» yy = x(:,2)

     yy =

     6
     1
     5

» z = x(1,:)

z =
     2     6     8
```

The colon operator is also useful in extracting smaller matrices from larger matrices. If the 4 x 3 matrix c is defined by

```
     c = [ -1  0  0
            1  1  0
            1 -1  0
            0  0  2 ]

» d1 = c(:,2:3)
```

creates the following 4 x 2 matrix:

```
d1 =

     0     0
     1     0
    -1     0
     0     2

» d2 = c(3:4,1:2)
```

creates a 2 x 2 matrix in which the rows are defined by the 3rd and 4th row of c and the columns are defined by the 1st and 2nd columns of the matrix, c.

```
d2 =

     1    -1
     0     0
```

Solving linear systems of equations with matrix inversion and division

If *A* is a square, nonsingular matrix, then the solution of the equation *Ax=b* is *x=inv(A)b*. Matlab implements this operation with the backslash operator.

Suppose we have a matrix A and a vector b of random numbers. We can generate these arrays with the following commands:

`>> rand`	returns a random number between 0 and 1.
`>> randn`	returns a random number selected from a normal distribution with a mean of 0 and variance of 1.
`>> rand(A)`	returns a matrix of size A of random numbers from a uniform distribution with mean of 0 and variance of 1.

```
>> A = rand(3,3)

A =

    0.9501    0.4860    0.4565
    0.2311    0.8913    0.0185
    0.6068    0.7621    0.8214

>> b = rand(3,1)

b =

    0.4447
    0.6154
    0.7919
```

The can then find the solution to this system with:

```
>> x = A\b
x =
   -0.0638
    0.6995
    0.3622
```

Thus A\b is (mathematically) equivalent to multiplying *b* on the left by `inv(A)`. If you try to do that, you will obtzain the same solution. Try it yourself typing `inv(A)*b`. Another example of how to solve a system of equations using the matrix inverse and the matrix division is given below. Consider the following system of three equations.

```
x1 - 4x2 + 3x3 = -7

3x1 +  x2 - 2x3 = 14

2x1 +  x2 +  x3 =  5
```

Note that you can write this system as Ax = b, where:

```
A = [  1  -4   3
       2   1  -2
       2   1   1]

x = [   x1
        x2
        x3]

b = [   -7
        14
        5]
```

Using 'matlab', define the two matrices for [a] and [b].

```
>>  a = [ 1 -4  3; 3  1 -2; 2  1  1];
>>  b = [ -7; 14; 5];
```

and find the solution vector, x, using the inverse:

```
>>  x = inv(a)*b

x =

    3.0000
    1.0000
   -2.0000
```

Or find the solution using the \ operator:

```
>>  x = a\b

x =

     3
     1
    -2
```

It is often useful, when entering a matrix, to suppress the display. this is done by ending the line with a semicolon.

```
» E = [1 2;3 4];
```

5. Conditionals and loops

Matlab has a standard if-elseif-else conditional. For example:

```
>> t = rand(1);
>> if t > 0.75
      s = 0;
   elseif t < 0.25
      s = 1;
   else
      s = 1-2*(t-0.25);
   end

>> t
t =
    0.7622

>> s
s =
     0
```

Thus the general form of the if statement is

```
if expr1
   statements
 elseif expr2
   statements
  .
  .
 else
   statements
end
```

Matlab provides two types of loops, a for-loop and a while-loop. A for-loop repeats the statements in the loop as the loop index takes on the values in a given row vector:

```
>> for i=[1,2,3,4]
      disp(i^2)      (the built-in function disp, displays its
argument.)
   end

     1
     4
     9
    16
```

The loop must also be terminated by end. This loop would more commonly be written as:

```
>> for i=1:4         (recall that 1:4 is the same as [1,2,3,4])
      disp(i^2)
   end
```

```
      1
      4
      9
     16
```

The while-loop repeats as long as the given expr is true:

```
>> x=1;
>> while 1+x > 1
      x = x/2;
   end

>> x
x =
   1.1102e-16
```

Thus, the general form of a while loop is

```
      while   relation
          statements
      end
```

The statements will be repeatedly executed as long as the relation remains true. Another example: for a given number a, the following will compute and display the smallest nonnegative integer n such that 2^n >=a. Let, for example a=4.

```
>> n = 0;
>> while  2^n < a
>>  n = n + 1;
>> end
>> n
n =

     2
```

6. Scripts and Functions

Matlab includes many standard functions. Indeed, all of the standard functions such as sin, cos, log, exp, sqrt, as well as many others. Commonly used constants such as pi are also incorporated into Matlab. Some examples are (a complete list is provided at the end of the tutorial):

```
» sin(pi/4)
ans =

    0.7071
```

Note that, as long as you don't assign a variable a specific operation or result, Matlab will store it in a temporary variable called "ans".

```
» cos(.5)^2+sin(.5)^2
 ans =
       1
```

```
» exp(1)
  ans =
      2.7183

» log(ans)+1
  ans =
       2
```

To determine the usage of any function, type help [function name] at the Matlab command window. For example:

```
» help sqrt

 SQRT   Square root.
    SQRT(X) is the square root of the elements of X. Complex
    results are produced if X is not positive.
    See also SQRTM.
```

Note that, when entering a command such as sqrt into matlab what you are really doing is running an m-file or a script.

What is an m-file or a script?

An m-file is a simple text file where you can place Matlab commands, so, it is simply a collection of Matlab commands in an m-file (a text file whose name ends in the extension ".m", e.g. sqrt.m). When the file is run, Matlab reads the commands and executes them exactly as it would if you had typed each command sequentially at the Matlab prompt. To make life easier, choose a name for your m-file that doesn't already exist. To see if a filename.m exists, type help filename at the Matlab prompt.

```
» help eva

eva.m not found.
```

The m-file must be located in one of the directories in which Matlab automatically looks for m-files. One of the directories in which Matlab always looks is the **current working directory**; the command **cd** identifies the current working directory, and **cd newdir** changes from the working directory to **newdir.**

For simple problems, entering your requests at the Matlab prompt is fast and efficient. However, as the number of commands increases typing the commands over and over at the Matlab prompt becomes tedious. M-files will be helpful and almost necessary in these cases. You actually use m-files to create your own programs.

How to create, save, open and run an m-file?

To create an m-file, choose **New** from the **File** menu and select **m-file**. This procedure brings up a text editor window in which you can enter Matlab commands. To save the m-file when you have typed your commands, simply go to the **File** menu and choose **Save as** (remember to save it with the '.m' extension). To open an existing m-file, go to the **File** menu and choose **Open** .

After the m-file is saved with the name filename.m in the Matlab folder or directory, you can execute the commands in the m-file by simply typing filename at the Matlab prompt. You can also use m-files to create your own functions.

You can either type commands directly into matlab, or put all of the commands that you will need together in an m-file, and just run the file. If you put all of your m-files in the same directory that you run matlab from, then matlab will always find them.

How to create your own function?

The new function must be given a filename with a '.m' extension. For example create a function that is called addition.m, which will add two numbers. The first line of the file should contain the syntax for this function in the form:

```
function [output1,output2,…outputn] = filename(input1,input2,…,inputn)
```

The inputs are what you have to give to the function, in this case the two variables you want to add, and the output will be the sum of the two variables. So, you open a new m-file, as explained above, and type the following:

```
function [var3] = addition(var1,var2)
```

The next few lines contain the text that will appear when the help addition command is evoked. For example, you can write: `%addition is a function that adds two numbers`

These lines are optional, but must be entered using % in front of each line in the same way that you include comments in an ordinary m-file. Finally, below the help text, the actual function with all of the commands is included. In this case, we would then have:

```
function [var3] = addition(var1,var2)
%addition is a function that adds two numbers
var3 = var1+var2;
```

If you save these three lines in a file called "addition.m" in the Matlab directory, then you can use it always by typing at the command line:

```
» y = addition(3,8)

y =
    11
```

Obviously, most functions will be more complex than the one demonstrated here. This example just shows what the basic form looks like. Try help function for more information.

7. Solving nonlinear problems

In addition to functions for numerical linear algebra, Matlab provides functions for the solution of a number of common problems, such as numerical integration, initial value problems in ordinary differential equations, root-finding, and optimization.

Polynomials

In Matlab, a polynomial is represented by a vector. To create a polynomial in Matlab, simply enter each coefficient of the polynomial into the vector in descending order. For instance, let's say you have the following polynomial:

$$s^4 + 3s^3 - 15s^2 - 2s + 9$$

To enter this into Matlab, just enter it as a vector in the following manner

```
» x = [1 3 -15 -2 9]

x =
      1 3   -15   -2   9
```

Matlab can interpret a vector of length n+1 as an nth order polynomial. Thus, if your polynomial is missing any coefficients, you must enter zeros in the appropriate place in the vector. For example,

$$s^4 + 1$$

would be represented in Matlab as:

```
» y = [1  0  0  0  1]
```

You can find the value of a polynomial using the polyval function. For example, to find the value of the above polynomial at s=2,

```
» z = polyval([1  0  0  0  1],2)

z =
      17
```

You can also extract the roots of a polynomial. This is useful when you have a high-order polynomial such as

$$s^4 + 3s^3 - 15s^2 - 2s + 9$$

Finding the roots would be as easy as entering the following command;

```
» roots([1  3 -15 -2  9])

ans =
               -5.5745
                2.5836
               -0.7951
                0.7860
```

Optimization commands

- `fzero root-finding (single variable)`

Let now **f** be a function. MATLAB function **fzero** computes a zero of the function **f** using user supplied initial guess of a zero sought. In the following example let **f(x) = cos(x) –** x. First we define a function **y = f1(x)** in a separate m-file which we call function1.m:

```
 function y = f1(x)
y = cos(x) - x;
```

Then, we can type:

```
» r = fzero('function1', 0.5)
r =
0.73908513321516
```

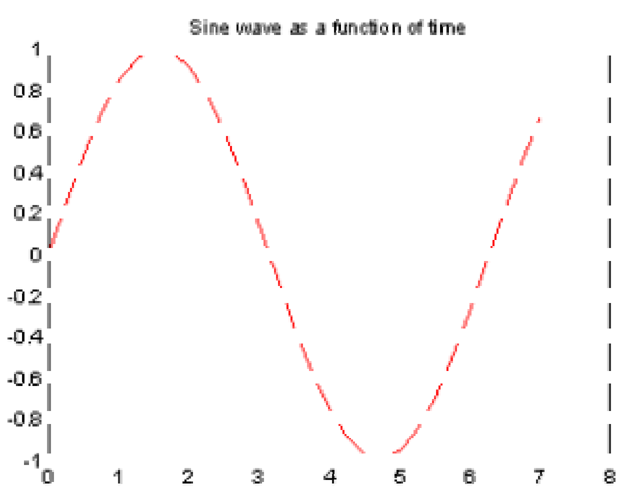
In the first line we tell matlab that f1 is a function and the second line defines the function. To compute its zero we use MATLAB function **fzero** and we have to give an initial guess (type help fzero for more information). Other useful built-in functions are:

- `fsolve` root-finding (several variables)
- `fmin` nonlinear minimization (single variable)
- `fmins` nonlinear minimization (several variables)

8. Plotting

It is easy to create plots in Matlab. Suppose you wanted to plot a sine wave as a function of time. First make a time vector (the semicolon after each statement tells Matlab we don't want to see all the values) and then compute the sin value at each time.

```
» t=0:0.25:7;
» y = sin(t);
» plot(t,y)
```



As you see. it is easy to create the needed vectors to graph a built-in function, since Matlab functions are *vectorized*. This means that if a built-in function such as sine is applied to a array, the effect is to create a new array of the same size whose entries are the function values of the entries of the original array.

The Plot function

One of the most important functions in Matlab is the plot function. Plot also happens to be one of the easiest functions to learn how to use. The basic format of the function is to enter the following command in the Matlab command window or into a m-file.

```
plot(x,y)
```

This command will plot the elements of vector x on the horizontal axis of a figure, and the elements of the vector y on the vertical axis of the figure. The default is that each time the plot command is issued, the current figure will be erased; we will discuss how to override this below. If we wanted to plot the simple, linear formula: y=3x, we could type the following (or create a m-file with the following lines of code):

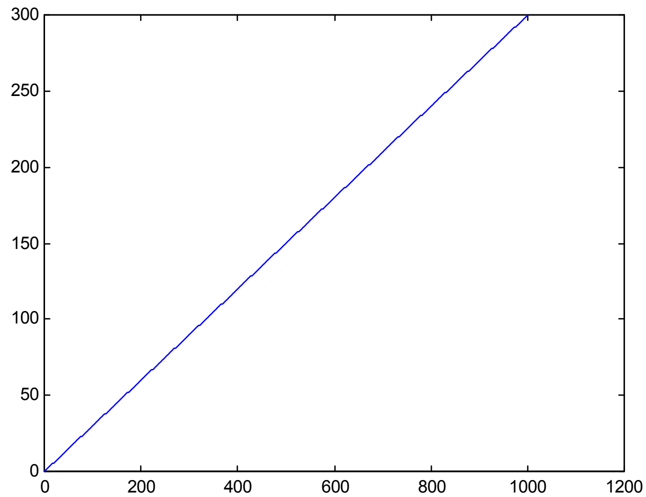
```
» x = 0:0.1:100;
» y = 3*x;
» plot(x,y)
```

which will generate the following plot,

One thing to keep in mind when using the plot command is that the vectors x and y must be the same length. The plot command can also be used with just one input vector. In this case, the vector 1:1:n will be used for the horizontal axis, where n is the length of y. So, if you type:

```
» plot(y)
```

you will get the following picture:

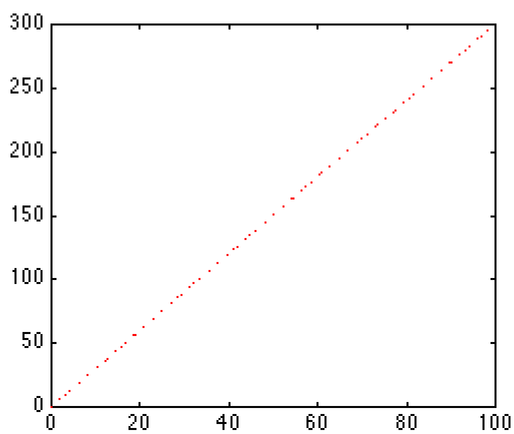


Plot aesthetics

The color and point marker can be changed on a plot by adding a third parameter (in single quotes) to the plot command. For example, to plot the above function as a red, dotted line, the m-file should be changed to:

```
» x = 0:0.1:100;
» y = 3*x;
» plot(x,y,'r:')
```

The plot now looks like:



The third input consists of one to three characters which specify a color and/or a point marker type. The list of colors and point markers is as follows:

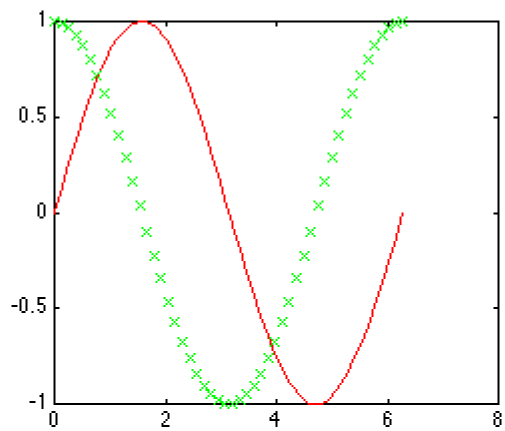
```
y     yellow        .     point
m     magenta       o     circle
c     cyan          x     x-mark
r     red           +     plus
```

```
g      green          -      solid
b      blue           *      star
w      white          :      dotted
k      black          -.     dashdot
                     --     dashed
```

You can plot more than one function on the same figure. Let's say you want to plot a sine wave and cosine wave on the same set of axes, using a different color and point marker for each. The following m-file could be used to do this:

```
» x = linspace(0,2*pi,50);
» y = sin(x);
» z = cos(x);
» plot(x,y,'r', x,z,'gx')
```

You will get the following plot of a sine wave and cosine wave, with the sine wave in a solid red line and the cosine wave in a green line made up of x's:



When plotting many things on the same graph it is useful to use the hold on and hold off commands. The same plot shown above could be generated using the following commands:

```
» x = linspace(0,2*pi,50);
» y = sin(x);
» plot(x,y,'r')
» z = cos(x);
» hold on
» plot(x,z,'gx')
» hold off
```

Always remember that if you use the hold on command, all plots from then on will be generated on one set of axes, without erasing the previous plot, until the hold off command is issued.

Subplotting

More than one plot can be put on the same figure using the subplot command. The subplot command allows you to separate the figure into as many plots as desired, and put them all in one figure. To use this command, the following line of code is entered into the Matlab command window or an m-file:

```
» subplot(m,n,p)
```

This command splits the figure into a matrix of m rows and n columns, thereby creating m*n plots on one figure. The p'th plot is selected as the currently active plot. For instance, suppose you want to see a sine wave, cosine wave, and tangent wave plotted on the same figure, but not on the same axis. The following m-file will accomplish this:

```
» x = linspace(0,2*pi,50);
» y = sin(x);
» z = cos(x);
» w = tan(x);
» subplot(2,2,1)
» plot(x,y)
» subplot(2,2,2)
» plot(x,z)
» subplot(2,2,3)
» plot(x,w)
```

There are only three plots, even though I created a 2 x 2 matrix of 4 subplots. Thus, you do not have to fill all of the subplots you have created, but Matlab will leave a spot for every position in the matrix. The subplots are arranged in the same manner as you would read a book. The first subplot is in the top left corner, the next is to its right.

One thing to note about the subplot command is that every plot command issued later will place the plot in whichever subplot position was last used, erasing the plot that was previously in it. For example, in the m-file above, if a plot command was issued later in the m-file, it would be plotted in the third position in the subplot, erasing the tangent plot. To solve this problem, a new figure should be specified (using figure). For example, two plots will be opened if you type:

```
» figure(1)
» plot(x,y)
» figure(2)
» plot(x,z)
```

Changing the axis

The axis command changes the axis of the plot shown, so only the part of the axis that is desirable is displayed. The axis command is used by entering the following command right after the plot command (or any command that has a plot as an output):

```
» axis([xmin, xmax, ymin, ymax])
```

For instance, suppose want to look at a plot of the function y=exp(5t)-1. If you enter the following into Matlab

```
» t=0:0.01:5;
» y=exp(5*t)-1;
» plot(t,y)
```

you should have the following plot:

To get a better idea of what is going on in this plot, let's look at the first second of this function. Enter the following command into the Matlab command window.

```
» axis([0, 1, 0, 50])
```

and you should get the following plot:

Now you can see more clearly what is going on as the function moves toward infinity. You can customize the axis to your needs. Type help axis for more information.

Adding text

Another thing that may be important for your plots is labeling. You can give your plot a title (with the title command), x-axis label (with the xlabel command), y-axis label (with the ylabel command), and put text on the actual plot. All of the above commands are issued after the actual plot command has been issued.

A title will be placed, centered, above the plot with the command: title('title string'). The x-axis label is issued with the following command: xlabel('x-axis string'). The y-axis label is issued with the following command: ylabel('y-axis string').

Furthermore, text can be put on the plot itself with the command gtext('textstring'), and then you just move the cross-hair to the desired location with the mouse, and click on the position you want the text placed.

Suppose you have a plot and you type:

```
» title('step response of something')
» xlabel('time (sec)')
» ylabel('position, velocity, or something like that')
» gtext('unnecessary labeling')
```

The text "unnecessary labeling" was placed right above the position, I clicked on. The plot should look like the following:

You may wish to save in a graphics file, a plot you have created. To do so, simply append to the 'print' command the name of the file. The command

```
>> print filename
```

will store the contents of the graphics window in the file titled 'filename.ps' in a format called postscript.

9. Basic Commands and Functions

- `max(x)` returns the largest entry of x, if x is a vector; see help max for the result when x is a *k*-dimensional array
- `min(x)` analogous to max
- `abs(x)` returns an array of the same size as x whose entries are the magnitudes of the entries of x
- `mean(x)` returns the mean value of the elements of a vector or if x is a matrix, returns a row vector whose elements are the mean value of the elements from each column of the matrix.
- `median(x)` same as mean(x), only returns the median value.
- `sum(x)` returns the sum of the elements of a vector or if x is a matrix, returns the sum of the elements from each respective column of the matrix
- `prod(x)` same as sum(x), only returns the product of elements.
- `std(x)` returns the standard deviation of the elements of a vector or if x is a matrix, a row vector whose elements are the standard deviations of each column of the matrix.
- `sort(x)` sorts the values in the vector x or the columns of a matrix and places them in ascending order. Note that this command will destroy any association that may exist between the elements in a row of matrix x.
- `size(A)` returns a **1x k** vector with the number of rows, columns, etc. of the *k*-dimensional array A
- `save fname` saves the current variables to the file named fname.mat
- `load fname` load the variables from the file named fname.mat
- `clear x` erases the matrix 'x' from your workspace
- `clear or clear all` erases ALL matrices from your workspace

Line continuation

Occasionally, a line is so long that it can not be expressed in the 80 spaces available on a line, in which case a line continuation is needed. In matlab, the ellipsis defining a line continuation is three successive periods, as in "...". For example:

```
>>  4 +  5  +  3 ...
      +  1  +  10  +  2 ...
      + 5
```

10. Suggestions

These are a few pointers about programming and programming in MATLAB in particular.

- Use the indented style. It makes the programs easier to read, the program syntax is easier to check, and it forces you to think in terms of building your programs in blocks.
- In MATLAB, try to avoid loops in your programs. MATLAB is optimized to run the built-in functions. The following two command sequences have the same effect:

  ```
  >> t = (0:0.001:1)';
  >> y=sin(t);
  ```

  and

  ```
  >> t = (0:0.001:1)';
  >> for i=1:length(t)
         y(i) = sin(t(i));
     end
  ```

  However, the explicit for-loop takes much longer as the vectorized sine function. Of course there will be occasions where you cannot avoid using the **for** loop especially when you are working with recursive problems. In this case, you should always assign an initial array of e.g. zeros that will be filled up as the loop progresses.

- Always supress any unecessary outputs with the semicolon (;). If you want to see the outputs as the programme runs, beware that the speed of execution will be significantly higher. When you write your m-file start it with a command **clear all;** this will clear the memory and improve the performance of the processor.
- Last, although it's good to write comments (using %) in your m-file, these increase the execution time, as the computer actually "reads" the line but does not execute it. So, here you face the trade-off of computing time versus readability of your code.

11. Built-in Functions and Help Topics

Built-in functions

This is a list of functions available in Matlab as of 1984, which should be taken as a quick reminder of the most basic tools available.

```
intro    <        chol     end      function lu       quit     sprintf
help     >        clc      eps      global   macro    qz       sqrt
demo     =        clear    error    grid     magic    rand     startup
[        &        clg      eval     hess     max      rcond    string
]        |        clock    exist    hold     memory   real     subplot
(        ~        conj     exit     home     mesh     relop    sum
)        abs      contour  exp      ident    meta     rem      svd
.        all      cos      expm     if       min      return   tan
,        ans      cumprod  eye      imag     nan      round    text
;        any      cumsum   feval    inf      nargin   save     title
%        acos     delete   fft      input    norm     schur    type
!        asin     det      filter   inv      ones     script   what
:        atan     diag     find     isnan    pack     semilogx while
'        atan2    diary    finite   keyboard pause    semilogy who
+        axis     dir      fix      load     pi       setstr   xlabel
-        balance  disp     floor    log      plot     shg      ylabel
*        break    echo     flops    loglog   polar    sign     zeros
\        casesen  eig      for      logop    prod     sin
/        ceil     else     format   ltifr    prtsc    size
^        chdir    elseif   fprintf  ltitr    qr       sort
```

```
acosh        demo         hankel       membrane     print        table1
angle        demolist     hds          menu         quad         table2
asinh        dft          hilb         meshdemo     quaddemo     tanh
atanh        diff         hist         meshdom      quadstep     tek
bar          eigmovie     histogram    mkpp         rank         tek4100
bench        ergo         hp2647       movies       rat          terminal
bessel       etime        humps        nademo       ratmovie     toeplitz
bessela      expm1        idft         nelder       readme       trace
besselh      expm2        ieee         neldstep     residue      translate
besseln      expm3        ifft         nnls         retro        tril
blanks       feval        ifft2        null         roots        triu
cdf2rdf      fft2         info         num2str      rot90        unmkpp
census       fftshift     inquire      ode23        rratref      vdpol
citoh        fitdemo      int2str      ode45        rratrefmovie versa
cla          fitfun       invhilb      odedemo      rref         vt100
compan       flipx        isempty      orth         rsf2csf      vt240
computer     flipy        kron         pinv         sc2dc        why
cond         funm         length       plotdemo     sg100        wow
conv         gallery      log10        poly         sg200        xterm
conv2        gamma        logm         polyfit      sinh         zerodemo
corr         getenv       logspace     polyline     spline       zeroin
cosh         ginput       matdemo      polymark     sqrtm
ctheorem     gpp          matlab       polyval      square
dc2sc        graphon      mean         polyvalm     std
deconv       hadamard     median       ppval        sun
```

```
addtwopi buttap   cov      fftdemo  freqz    kaiser   specplot
bartlett butter   decimate filtdemo fstab    numf     spectrum
bilinear chebap   denf     fir1     hamming  readme2  triang
blackman chebwin  detrend  fir2     hanning  remez    xcorr
boxcar   cheby    eqnerr2  freqs    interp   remezdd  xcorr2
yulewalk
```

```
>> help

HELP topics:

matlab/general       -  General purpose commands.
matlab/ops           -  Operators and special characters.
matlab/lang          -  Language constructs and debugging.
matlab/elmat         -  Elementary matrices and matrix manipulation.
matlab/specmat       -  Specialized matrices.
matlab/elfun         -  Elementary math functions.
matlab/specfun       -  Specialized math functions.
matlab/matfun        -  Matrix functions - numerical linear algebra.
matlab/datafun       -  Data analysis and Fourier transform functions.
matlab/polyfun       -  Polynomial and interpolation functions.
matlab/funfun        -  Function functions - nonlinear numerical methods.
matlab/sparfun       -  Sparse matrix functions.
matlab/plotxy        -  Two dimensional graphics.
matlab/plotxyz       -  Three dimensional graphics.
matlab/graphics      -  General purpose graphics functions.
matlab/color         -  Color control and lighting model functions.
matlab/sounds        -  Sound processing functions.
matlab/strfun        -  Character string functions.
matlab/iofun         -  Low-level file I/O functions.
matlab/demos         -  The MATLAB Expo and other demonstrations.
toolbox/chem         -  Chemometrics Toolbox
toolbox/control      -  Control System Toolbox.
fdident/fdident      -  Frequency Domain System Identification Toolbox
fdident/fddemos      -  Demonstrations for the FDIDENT Toolbox
toolbox/hispec       -  Hi-Spec Toolbox
toolbox/ident        -  System Identification Toolbox.
toolbox/images       -  Image Processing Toolbox.
toolbox/local        -  Local function library.
toolbox/mmle3        -  MMLE3 Identification Toolbox.
mpc/mpccmds          -  Model Predictive Control Toolbox
mpc/mpcdemos         -  Model Predictive Control Toolbox
mutools/commands     -  Mu-Analysis and Synthesis Toolbox.: Commands directory
mutools/subs         -  Mu-Analysis and Synthesis Toolbox -- Supplement
toolbox/ncd          -  Nonlinear Control Design Toolbox.
nnet/nnet            -  Neural Network Toolbox.
nnet/nndemos         -  Neural Network Demonstrations and Applications.
toolbox/optim        -  Optimization Toolbox.
toolbox/robust       -  Robust Control Toolbox.
toolbox/signal       -  Signal Processing Toolbox.
toolbox/splines      -  Spline Toolbox.
toolbox/stats        -  Statistics Toolbox.
toolbox/symbolic     -  Symbolic Math Toolbox.
toolbox/wavbox       - (No table of contents file)
simulink/simulink    -  SIMULINK model analysis and construction functions.
simulink/blocks      -  SIMULINK block library.
simulink/simdemos    -  SIMULINK demonstrations and samples.
toolbox/codegen      -  Real-Time Workshop
```

12. Useful Sites on the Web

-The MathWorks Web site: http://www.mathworks.com/

-Matlab Educational Sites: http://www.eece.maine.edu/mm/matweb.html

-Some Matlab Links: http://math.uc.edu/~kingjt/matlab_lnk.html

-Matlab resources on the web:

http://www.eeng.brad.ac.uk/help/.packlangtool/.maths/.matlab/.resource.html

-Online Matlab Tutorials

http://mechanical.poly.edu/faculty/vkapila/matlabtutor.htm

-One of the best tutorials I found on the web:

http://www.math.siu.edu/matlab/tutorials.html

13. Practice Exercises

1. Determine the size for the following vectors and matrices. Enter them in Matlab and check your results using the 'whos' statement.

```
a = [1,0,0,0,0,1]

b = [2;4;6;10]

c = [5  3  5; 6  2  -3]


e = [3  5  10  0;  0  0 ...
     0  3;  3  9  9  8  ]

t = [4  24  9]

q = [t 0 t]
```

2. Define the 5 x 4 matrix, g.

```
    g = [ 0.6  1.5  2.3 -0.5
          8.2  0.5 -0.1 -2.0
          5.7  8.2  9.0  1.5
          0.5  0.5  2.4  0.5
          1.2 -2.3 -4.5  0.5 ]
```

Determine the content and size of the following matrices and check your results for content and size using matlab.

```
a = g(:,2)

b = g(4,:)

c = [10:15]

d = [4:9;1:6]

e = [-5,5]

f= [1.0:-.2:0.0]
```

```
 t1 = g(4:5,1:3)
```

3. Find the solution to

```
2x1 +  x2 - 4x3 + 6x4 + 3x5 - 2x6 = 16

-x1 + 2x2 + 3x3 + 5x4 - 2x5       = -7

 x1 - 2x2 - 5x3 + 3x4 + 2x5 +  x6 =  1

4x1 + 3x2 - 2x3 + 2x4       +  x6 = -1

3x1 +  x2 -  x3 + 4x4 + 3x5 + 6x6 = -11

5x1 + 2x2 - 2x3 + 3x4 +  x5 +  x6 =  5
```

using the matrix inverse and matrix division.

4.Create a ten-dimensional row vector whose all components are equal 2.

5. Let x = [2 5 1 6].

   a. Add 16 to each element
   b. Add 3 to just the odd elements
   c. Compute the square root of each element
   d. Compute the square of each element

6. Let x = [3 2 6 8]' and y = [4 1 3 5]'

   a. Add the sum of the elements in x to y
   b. Raise each element of x to the power specified by the corresponding element in y.
   c. Multiply each element in x by the corresponding element in y, calling the result "z".
   d. Add up the elements in z

7. Evaluate the following MATLAB expressions by hand and use MATLAB to check the answers

```
a. 2 / 2 * 3
b. 6 - 2 / 5 + 7 ^ 2 - 1
c. 10 / 2 \ 5 - 3 + 2 * 4
d. 3 ^ 2 / 4
e. 3 ^ 2 ^ 2
```

8. Create a vector x with the elements ...

   a. 2, 4, 6, 8, ...
   b. 10, 8, 6, 4, 2, 0, -2, -4
   c. 1, 1/2, 1/3, 1/4, 1/5, ...
   d. 0, 1/2, 2/3, 3/4, 4/5, ...

9. Plot the functions x, x^3, e^x and e^{x2} over the interval 0 < x < 4. Put a title to each function and add some text inside the graphs.

10. Make a good plot (i.e., a non-choppy plot) of the function

```
f(x) = sin(1/x)   for 0.01 < x < 0.1.
```

11. Given x = [3 1 5 7 9 2 6], explain what the following commands "mean" by summarizing the net result of the command.

```
a. x(3)
b. x(1:7)
c. x(1:end)
d. x(1:end-1)
e. x(6:-2:1)
g. sum(x)
```

12. Given the array A = [2 4 1 ; 6 7 2 ; 3 5 9], provide the commands needed to

  a. assign the first row of A to a vector called x1
  b. assign the last 2 rows of A to an array called y
  c. compute the sum over the columns of A
  d. compute the sum over the rows of A

13. Given the arrays x = [1 4 8], y = [2 1 5] and A = [3 1 6 ; 5 2 7], determine which of the following statements will correctly execute and provide the result. If the command will not correctly execute, state why it will not. Using the command **whos** may be helpful here.

```
a. x + y
b. x + A
c. x' + y
d. A - [x' y']
e. [x ; y']
f. [x ; y]
g. A - 3
```

14. Given the array A = [2 7 9 7 ; 3 1 5 6 ; 8 1 2 5], explain the results of the following commands:

```
a. A'
b. A(:,[1 4])
e. A(:)
h. [A A(end,:)]
i. A(1:3,:)
j. [A ; A(1:2,:)]
m. sum(A,2)
```

15. Given the array A from problem 4, above, provide the command that will

  a. assign the even-numbered columns of A to an array called B
  b. assign the odd-numbered rows to an array called C
  c. convert A into a 4-by-3 array
  e. compute the square-root of each element of A

16. Provide the right answers and use MATLAB to check them.

```
1.   if n > 1            a. n = 7   m = ?
```

```
      m = n+1          b. n = 0   m = ?
   else                c. n = -10 m = ?
      m = n - 1
   end

2. if z < 5            a. z = 1    w = ?
      w = 2*z          b. z = 9    w = ?
   elseif z < 10       c. z = 60   w = ?
      w = 9 - z        d. z = 200  w = ?
   elseif z < 100
      w = sqrt(z)
   else
      w = z
   end

3. if T < 30           a. T = 50    h = ?
      h = 2*T + 1      b. T = 15    h = ?
   elseif T < 10       c. T = 0     h = ?
      h = T - 2
   else
      h = 0
   end

4. if 0 < x < 10           a. x = -1   y = ?
      y = 4*x              b. x = 5    y = ?
   elseif 10 < x < 40      c. x = 30   y = ?
      y = 10*x             d. x = 100  y = ?
   else
      y = 500
   end
```

17. Given the vector x = [1 8 3 9 0 1], create a short set of commands that will (use loops for this)

   a. Add up the values of the elements (Check the result with **sum**.)
   b. computes the sine of the given x-values

18. Create an M-by-N array of random numbers (use **rand**). Move through the array, element by element, and set any value that is less than 0.2 to 0 and any value that is greater than (or equal to) 0.2 to 1. (use loops for this)

19. Given x = [4 1 6] and y = [6 2 7], compute the following arrays

   a. $a_{ij} = x_i y_j$ (i.e, the element a(1,1) will be x1*y1)
   b. $b_{ij} = x_i / y_j$
   c. $c_i = x_i y_i$, then add up the elements of c.

JOURNAL OF COMPARATIVE ECONOMICS **23,** 192–208 (1996)
ARTICLE NO. 0054

Estimating Economic Effects of Political Movements in China[1]

YUM K. KWAN

Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

AND

GREGORY C. CHOW

Princeton University, Princeton, New Jersey 08544

Received November 30, 1995; revised April 16, 1996

Kwan, Yum K., and Chow, Gregory C.— Estimating Economic Effects of Political Movements in China

To measure the economic effects of political movements in China a simple econometric model is constructed. Investment is determined by a central planner maximizing a multiperiod objective function. Political events are modeled by exogenous changes in the shocks to productivity and to investment which affect the time paths of major economic variables. Effects of the events are measured by comparing the time paths generated by the model with and without the changes in the shocks. Without the Great Leap output and consumption per capita would have been 2.0 times as great in 1993, without the Cultural Revolution, 1.2 times as great. *J. Comp. Econom.,* October 1996, **23**(2), pp. 192–208. Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong; and Princeton University, Princeton, New Jersey 08544. © 1996 Academic Press, Inc.

Journal of Economic Literature Classification Numbers: E65, O47, P24, P26.

1. INTRODUCTION

What were the economic effects of the Great Leap Forward Movement in 1958–1962 and the Cultural Revolution in 1966–1969 in China? In other words, if these two events had not occurred, what would have been

[1] Ordering of authors was determined by randomization. The authors thank two referees and the Editor for helpful comments which are incorporated in the revision of this paper.

0147-5967/96 $18.00

the time paths of the major economic variables such as consumption, real output, and capital stock in the years following 1958? This is an interesting question in economic history. To answer this question one has to compare the historical time paths of these variables with the paths that would have prevailed absent the above events. We first construct an econometric model to explain the growth of the Chinese economy that incorporates the shocks from these two political events. Then the shocks are removed and the hypothetical time paths of the major economic variables are generated from the model. Comparing the hypothetical time paths with the time paths incorporating the shocks provides an answer to our question.

The econometric model has only one sector and includes aggregate output, consumption, investment, physical capital stock and total labor force as major variables. Aggregate output is produced by physical capital and labor according to a Cobb–Douglas production function. Output is divided into consumption and net investment, the latter measured by accumulation in Chinese official statistics. Capital stock increases by the flow of investment. To determine investment, we assume that actual investment equals planned investment plus an error. Planned investment is determined by the assumption that a central planner maximizes a multiperiod objective function with consumption per laborer as argument. The error may be affected by political events. The logarithm of total factor productivity follows a random walk with drift in normal years. In abnormal years such as during the Great Leap and the Cultural Revolution, the residual of the random walk process can also be affected. Thus the effects of political events are modeled by changes in the error of the investment function and in the residual of the random walk process for productivity. Having estimated such a model, one can remove the changes in order to measure the economic effects of the two political events. Section 2 specifies the model and the data. Section 3 presents the method of estimation and the parameter estimates. Section 4 reports on the time paths of major variables obtained by simulating the model absent the shocks from the two political events and provides measures of economic losses attributable to them. Section 5 concludes.

2. MODEL AND DATA

The econometric model consists of four equations. A Cobb–Douglas production function determines aggregate real output Q by physical capital stock K and labor L with constant return to scale. Denoting Q/L and K/L by q and k, respectively, and net investment per laborer by i, we can write the production function, the output identity, the capital accumulation equation, and the equation explaining total factor productivity A as

$$q_t = A_t k_t^{1-\alpha} \tag{1}$$

$$q_t = c_t + i_t \tag{2}$$

$$k_{t+1} = k_t + i_t \tag{3}$$

$$\ln A_{t+1} = \gamma + \ln A_t + \eta_{t+1}, \tag{4}$$

where η is a random shock to the logarithm of total factor productivity A. Note that the capital accumulation equation is obtained by dividing the original identity in aggregate variables by labor L in two adjacent periods and is therefore only an approximation.

The data for aggregate output Q are national income used (*Statistical Yearbook of China* 1994, abbreviated SYB, p. 40) divided by the implicit price deflator of national income. The price deflator is the ratio of national income in current prices (SYB, p. 33; measured in 100 million yuan) to national income in 1952 prices; the latter equals 589 (national income in 1952 in 100 million yuan) times the index of real national income (SYB, p. 34; =100 in 1952) divided by 100. In Chinese official statistics, national income used equals consumption plus accumulation (net investment) in current prices. In our model this identity is assumed to hold in constant prices. We have estimated real national income used Q, real consumption C, and real net investment I by dividing their current values (SYB, p. 40) by the above price deflator. Labor L is total labor force (SYB, p. 88). Given $K = 2213$ (100 million yuan) in 1952 (an estimate from Chow (1993b, p. 821)), we estimate k in 1952 by K/L and k in later years by Eq. (3).

We assume that the Chinese economy evolves as if there were a central planner who, knowing the parameters of the model as we have specified, tries to maximize the following objective function at the beginning of each period t,

$$E_t \sum_{i=1}^{\infty} \beta^{t+i} \log c_{t+i}, \tag{5}$$

subject to the constraints in (1)–(4). This dynamic optimization problem can be solved by defining the control variable as either consumption per laborer c, or investment per laborer i, or even next-period capital stock, as they are related by the identities (2) and (3). This maximization assumption might be questioned. A critic might argue that economic planners in China are not so rational as to have a specific objective function. She would say, just look at what happened to rational economic planning during the Great Leap and the Cultural Revolution. Our response is that during these abnormal periods there were exogenous shocks to the production and investment processes in China, caused to a large extent by the behavior of Chairman Mao, which the economic planners could not control. However, given these shocks, the planners still attempted in each period to maximize the above objective function from that period onward.

Among the possible shortcomings of this model are the treatment of technology, population, and labor force as exogenous and the failure to incorporate possible effects of outcomes on human capital formation. Despite these possible shortcomings, we believe that the present study is an important step toward measuring the economic effects of the two major political events and can serve as a benchmark for incorporating other important effects in future research.

3. STATISTICAL ESTIMATION

As discussed in the last section, the observed Chinese time series data on output, consumption, and capital are interpreted as the outcome of a dynamic optimization process. The solution to the dynamic optimization problem will depend on the parameters (α, β, γ) and the process governing the evolution of productivity. When we estimate the parameters by the method of maximum likelihood, we are in fact searching for a set of parameters for which the solution to the dynamic optimization problem and the observed series are as close as possible. A dynamic optimization problem is thus embedded within each evaluation of the likelihood function. More precisely, calculating the likelihood value for a given parameter setting proceeds in two stages. First, an optimal decision function for investment is determined by assuming that the central planner in China maximizes the objective function (5) subject to the constraints of the model (1)–(4) at each period t. Second, the optimal decision function is combined with the original model to form an econometric model for which the likelihood value can be calculated.

The dynamic optimization problem as stated in (1)–(5) can be converted into an equivalent version involving only stationary processes. The idea is to detrend all variables along their balanced growth paths. After detrending, the model is stationary and econometric problems associated with the unit root in Eq. (4) will be avoided. Define

$$z_t = A_t^{1/\alpha}, \qquad \bar{k}_{t+1} = k_{t+1}/z_t, \qquad \bar{c}_t = c_t/z_t, \qquad \bar{z}_t = z_t/z_{t-1}. \tag{6}$$

Replacing i by $q - c$ and q by the production function, we can write the capital accumulation equation as

$$k_{t+1} = k_t + A_t k_t^{1-\alpha} - c_t,$$

implying

$$k_{t+1}/z_t = (k_t/z_{t-1})z_{t-1}/z_t + k_t^{1-\alpha} z_t^{\alpha-1} - c_t/z_t,$$

or, in terms of the detrended variables defined in (6),

$$\bar{k}_{t+1} = \bar{k}_t \bar{z}_t^{-1} + \bar{k}_t^{1-\alpha} \bar{z}_t^{\alpha-1} - \bar{c}_t. \tag{7}$$

Similarly the productivity Eq. (4) can be written as

$$\ln \bar{z}_t = \mu + \epsilon_t, \tag{8}$$

where

$$\mu = \gamma/\alpha, \qquad \epsilon_t = \eta_t/\alpha.$$

Since z_t is exogenous, we may replace the objective function (5) by

$$E_t \sum_{i=1}^{\infty} \beta^{t+i} \ln \bar{c}_{t+i}. \tag{9}$$

Maximizing (9) subject to (7)–(8) is equivalent to the non-stationary version in (1)–(5). We approach the dynamic optimization problem by first substituting (7) into (9) to eliminate the detrended consumption variable, and then define the control variable to be $\ln \bar{k}_{t+1}$, and two state variables $\ln \bar{z}_t$ and $\ln \bar{k}_t$. With state and control so defined, we obtain numerically an approximate solution in the form of a log-linear first-order difference equation:

$$\ln \bar{k}_t = g + G_1 \ln \bar{z}_{t-1} + G_2 \ln \bar{k}_{t-1}. \tag{10}$$

The coefficients (g, G_1, G_2) may be regarded as reduced form parameters, as they are implicit functions of the three structural parameters (α, β, γ). The solution procedure and numerical algorithm can be found in the Appendix. It is well known that if the capital stock depreciates completely in one year, i.e., if $A_t k_t^{1-\alpha}$ replaces $k_t + A_t k_t^{1-\alpha}$ in the equation determining k_{t+1}, a closed form solution for (10) exists. However, this assumption is a poor approximation to reality and a numerical solution serves our purpose just as well.

Having derived the planned capital stock as described by (10), we allow actual capital stock to differ from planned capital by an error e due partly to failure of the planner to execute the plan and partly to failure of our simple model to capture the complicated economy completely. The econometric model to be estimated consists of two equations, an equation for $\ln \bar{z}_t$ and an equation for $\ln \bar{k}_t$, which can be written as a system of two regression equations,

$$y_t = \Gamma x_t + \xi_t, \tag{11}$$

where $y_t = (\ln \bar{z}_t, \ln \bar{k}_t)'$, $x_t = (1, \ln \bar{z}_{t-1}, \ln \bar{k}_{t-1})'$ and

$$\Gamma = \begin{bmatrix} \mu & 0 & 0 \\ g & G_1 & G_2 \end{bmatrix} \qquad \xi_t = \begin{bmatrix} \epsilon_t \\ e_t \end{bmatrix}. \tag{12}$$

With n observations (11) can be stacked up as

$$Y = X\Gamma + \Xi \tag{13}$$

with the transpose of (11) being the tth row of (13).

Assuming normal and serially uncorrelated residuals, and ξ_t having covariance matrix Σ, we can use the well-known concentrated log-likelihood function (see Chow (1983), pp. 170–171)

$$\ln L = \text{const} - (n/2)\ln|n^{-1}(Y - X\Gamma)'(Y - X\Gamma)| \tag{14}$$

and the maximum likelihood estimate of Σ is given by

$$\hat{\Sigma} = n^{-1}(Y - X\Gamma)'(Y - X\Gamma). \tag{15}$$

The observed variables are $\ln q_t$ and $\ln k_t$, with $\ln z_t$ defined as $[\ln q_t - (1 - \alpha)\ln k_t]/\alpha$. Substituting this definition for $\bar{z}_t = z_t/z_{t-1}$ in Eqs. (8) and (10), we have

$$\ln q_t - (1 - \alpha)\ln k_t = \gamma + \ln q_{t-1} - (1 - \alpha)\ln k_{t-1} + \eta_t \tag{8$'$}$$

$$\begin{aligned}\ln k_t = g + [\ln q_{t-1} - (1-\alpha)\ln k_{t-1}]/\alpha + G_1[\Delta\ln q_{t-1} - (1-\alpha)\Delta \ln k_{t-1}]/\alpha \\ + G_2[\ln k_{t-1} - (\ln q_{t-2} - (1 - \alpha)\ln k_{t-2})/\alpha] + e_t.\end{aligned} \tag{10$'$}$$

The Jacobian of the transformation from $\eta_t = \alpha\epsilon_t$ and e_t to $\ln q_t$ and $\ln k_t$ is

$$J = \begin{vmatrix} 1 & -(1 - \alpha) \\ 0 & 1 \end{vmatrix} = 1$$

which is implicit in the likelihood (14).

To calculate likelihood value for the parameters (α, β, μ) we use these parameters and the data on output and capital to compute z from the production function, $\bar{z}_t$ and $\bar{k}_t$ from Eq. (6), and the coefficients in Γ using Eq. (10). Thus the likelihood function (14) can be computed from the parameters and the data. We maximize the likelihood function in a sequential manner, i.e., $\max_\alpha$-$\max_{\beta,\mu}\ln L(\alpha, \beta, \mu)$. The maximization with respect to (β, μ) is performed by the MAXLIK package in GAUSS, and the linesearch in α is done by Brent's method (see Press et al. (1992, pp. 402–405)). The point estimate and standard error of γ can be recovered from that of α and μ via (8). To make sure that we have indeed located the global maximum, we have also used the simulated annealing algorithm as implemented by Goffe et al. (1991) to maximize the likelihood function. The sample period is from 1954 to 1993.

The maximum likelihood estimates of (α, β, γ), with standard errors given in parentheses, are

$$\begin{aligned}&(\hat{\alpha}, \hat{\beta}, \hat{\gamma}) = [0.7495\ (0.0108),\ 0.9999\ (0.0001),\ 0.0218\ (0.0025)] \\ &\text{mean log likelihood} = 6.6120, \text{ sample size} = 40.\end{aligned} \tag{16}$$

The estimate 0.7495 for labor elasticity of production is reasonable. It is

TABLE 1

Parameter Estimates for Fixed Values of α

α	$\hat{\beta}$	$\hat{\gamma}$	Mean log likelihood
0.4	0.9627 (0.0050)	0.0046 (0.0011)	5.9754
0.5	0.9715 (0.0037)	0.0083 (0.0017)	6.2012
0.6	0.9817 (0.0024)	0.0132 (0.0024)	6.3869
0.7	0.9940 (0.0015)	0.0194 (0.0033)	6.5456

somewhat higher than the estimate of about 0.4 reported in Chow (1993b, especially Table VII), but the latter study uses a deterministic trend for log total factor productivity and a sample period ending in 1980 whereas the current estimate is based on a stochastic trend and a sample period extending to 1993. The estimate 0.9999 for the annual discount factor is also reasonable in view of the high value which Chinese planners are supposed to place on future consumption or current investment at the expense of current consumption. This parameter is considered difficult to estimate statistically and is often imposed *a priori* in empirical studies of real business cycles in the United States. The positive drift of log total factor productivity of 0.0218 is also reasonable as the sample includes the post-reform years 1978–1993. It is consistent with Chow (1993b), which found no positive deterministic trend in total factor productivity during the sample period from 1952 to 1980 but a positive trend from 1979 on. Unlike Chow (1993b), the present study not only extends the sample period to 1993 but in estimating model parameters does not exclude any observations that are considered abnormal. This extension is possible partly because a stochastic trend is used for log total productivity rather than a linear deterministic trend as in Chow (1993b).

For sensitivity analysis we present in Table 1 estimates for the remaining two parameters when the labor elasticity parameter is fixed *a priori* at other values sometimes chosen in growth accounting exercises (see, e.g., Li et al. (1995)).

4. MEASURING THE EFFECTS OF TWO POLITICAL EVENTS

To estimate the economic effects of the Great Leap Forward alone we change the estimated residuals of the two reduced form equations in the years 1958–1962 to the mean values of the corresponding residuals in the remaining years; see Figs. 1 and 2. Columns 2 and 3 of Table 2 present actual output per laborer q_t, which can be generated by our model if the estimated residuals are used in the two equations, and simulated output q_t^*, which is generated

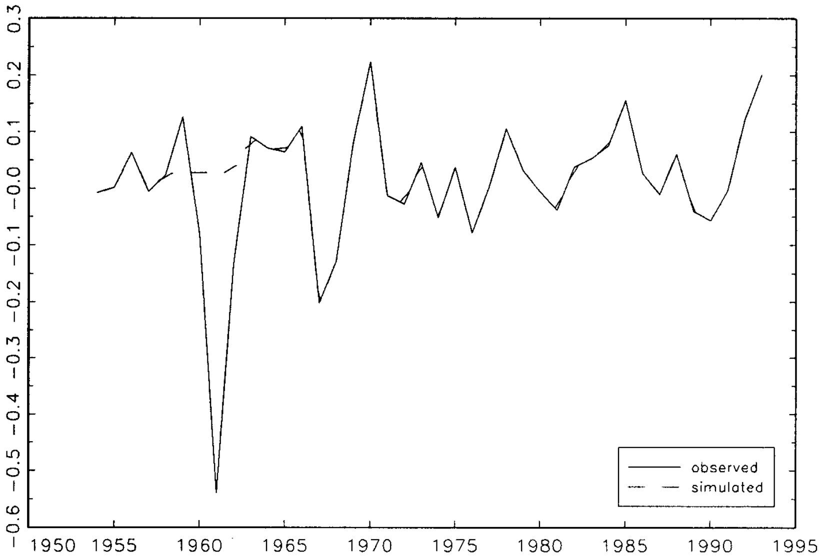


FIG. 1. Observed and simulated residual 1.

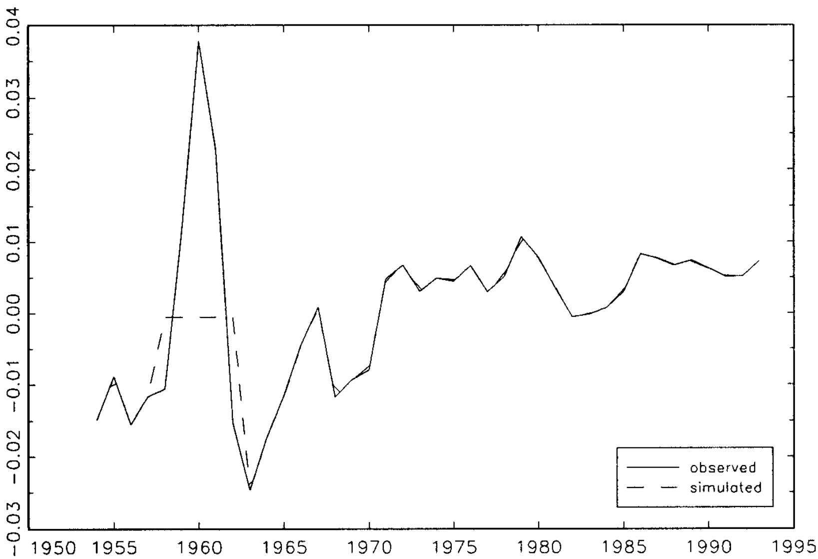


FIG. 2. Observed and simulated residual 2.

TABLE 2

GREAT LEAP FORWARD EFFECT

	Output		Consumption		Capital stock		Log productivity	
Year	Observed	Simulated	Observed	Simulated	Observed	Simulated	Observed	Simulated
1952	2.9283	2.9283	2.3011	2.3011	10.676	10.676	0.48132	0.48132
1953	3.2227	3.2227	2.4780	2.4780	11.303	11.303	0.56285	0.56285
1954	3.3276	3.3276	2.4794	2.4794	12.048	12.048	0.57888	0.57888
1955	3.4661	3.4661	2.6715	2.6715	12.896	12.896	0.60262	0.60262
1956	3.7717	3.7717	2.8500	2.8500	13.691	13.691	0.67214	0.67214
1957	3.9038	3.9038	2.9310	2.7747	14.612	14.612	0.69025	0.69025
1958	4.1304	4.1525	2.7289	2.9482	15.585	15.741	0.73053	0.73336
1959	4.7393	4.4162	2.6635	3.1322	16.986	16.946	0.84648	0.77647
1960	4.6947	4.6959	2.8339	3.3274	19.062	18.230	0.80816	0.81959
1961	3.2774	4.9924	2.6465	3.5344	20.923	19.598	0.42542	0.86270
1962	3.0530	5.3069	2.7342	4.2926	21.554	21.056	0.34707	0.90581
1963	3.3543	5.8781	2.7680	4.5729	21.873	22.070	0.43750	0.99625
1964	3.6400	6.4285	2.8314	4.8570	22.459	23.376	0.51262	1.0714
1965	3.9385	7.0076	2.8712	5.1104	23.268	24.947	0.58258	1.1413
1966	4.4182	7.9173	3.0654	5.6422	24.335	26.844	0.68628	1.2450
1967	3.9337	7.0975	3.0963	5.3530	25.688	29.119	0.55660	1.1153
1968	3.6809	6.6849	2.9024	4.9747	26.525	30.864	0.48213	1.0409
1969	4.0273	7.3599	3.0919	5.4272	27.303	32.574	0.56482	1.1236
1970	4.9087	9.0246	3.2916	6.2019	28.239	34.507	0.75429	1.3130
1971	5.0405	9.3203	3.3235	6.2964	29.856	37.329	0.76684	1.3256
1972	5.1180	9.5159	3.5018	6.5478	31.573	40.353	0.76810	1.3268
1973	5.4831	10.249	3.6789	6.9632	33.189	43.321	0.82451	1.3833
1974	5.4627	10.262	3.6958	6.9540	34.993	46.607	0.80751	1.3663
1975	5.8133	10.974	3.8447	7.3186	36.760	49.915	0.85738	1.4161
1976	5.6731	10.759	3.9225	7.3361	38.729	53.570	0.81990	1.3786

TABLE 2—*Continued*

Year	Output		Consumption		Capital stock		Log productivity	
	Observed	Simulated	Observed	Simulated	Observed	Simulated	Observed	Simulated
1977	5.8764	11.194	3.9762	7.4969	40.479	56.993	0.84404	1.4028
1978	6.5737	12.576	4.1718	8.0845	42.379	60.690	0.94469	1.5034
1979	6.9773	13.402	4.5635	8.8103	44.781	65.181	0.99046	1.5492
1980	7.1944	13.873	4.9267	9.4212	47.195	69.773	1.0080	1.5667
1981	7.2277	13.990	5.1806	9.8049	49.463	74.225	1.0008	1.5596
1982	7.6748	14.908	5.4636	10.415	51.510	78.410	1.0507	1.6094
1983	8.2453	16.072	5.7936	11.144	53.721	82.903	1.1119	1.6706
1984	9.0219	17.643	6.1797	12.032	56.173	87.831	1.1907	1.7494
1985	10.490	20.579	6.8177	13.562	59.015	93.442	1.3291	1.8878
1986	11.107	21.854	7.2579	14.450	62.687	100.46	1.3711	1.9298
1987	11.438	22.570	7.5423	14.988	66.536	107.86	1.3855	1.9443
1988	12.408	24.554	8.1293	16.238	70.431	115.44	1.4528	2.0115
1989	12.492	24.785	8.2737	16.466	74.710	123.76	1.4447	2.0034
1990	12.409	24.684	8.3370	16.514	78.928	132.08	1.4243	1.9830
1991	12.806	25.536	8.6106	17.069	83.000	140.25	1.4432	2.0019
1992	14.512	29.005	9.5145	19.074	87.196	148.72	1.5559	2.1147
1993	17.491	35.036			92.194	158.65	1.7286	2.2874
Mean	6.7323	12.244	4.4410	7.9904	39.606	57.273	0.90151	1.3469
Std dev	3.6746	8.1727	2.0767	4.7878	22.816	42.633	0.35652	0.47905

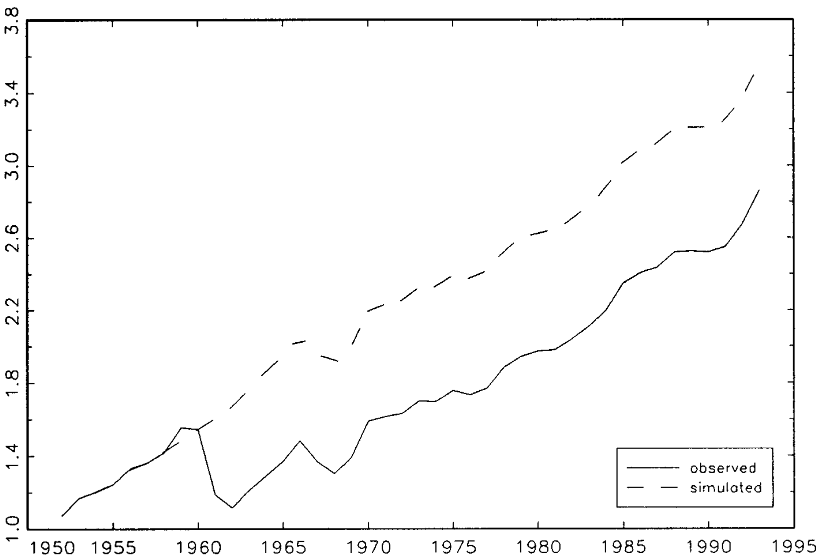


FIG. 3. Observed and simulated output (in log).

by our model if the estimated residuals in the years 1958–1962 are changed to the mean values of the remaining years. The remaining columns of Table 2 are the corresponding actual and simulated series for consumption, capital stock, and log productivity.

From Table 2 and Fig. 3 we observe that simulated output, which would have obtained absent the Great Leap, is about 2 times actual output in 1993. This result is derived from two sources. First, simulated total factor productivity in 1993 is about 0.56 higher than the actual value in logarithm, or about 1.74 times the actual value. Second, simulated capital stock is higher (see Fig. 5), and capital per laborer in 1993 is 1.72 times the actual value, as can be readily computed from the relevant entries in Table 2. According to our model and commonly used models of real business cycles for the United States economy, shifts in productivity due to shocks are permanent. Observe in Table 2 and Fig. 6 that simulated log productivity in 1962 is 0.9058, or 0.5587 higher than actual log productivity. The last figure equals 2.2874 − 1.7286, the difference between simulated and actual log productivity in 1993. Such a parallel shift in log productivity due to the Great Leap is clearly shown in Fig. 6. This is a characteristic of our model as Eq. (4) has a unit root, which implies a permanent shift in total factor productivity when its residual changes. The permanent shift in productivity in turn implies that log output and log consumption (see Fig. 3 and 5) will also undergo a permanent level shift. There is no effect on the steady state growth rate of each variable.

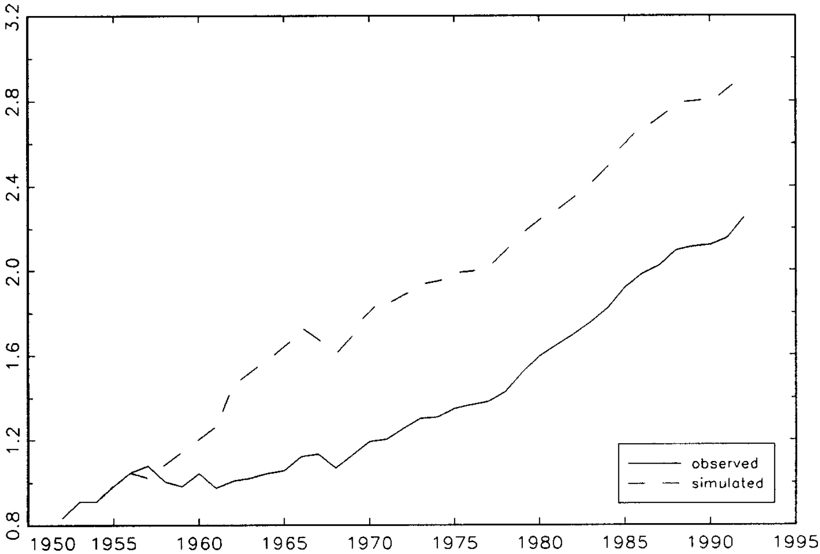


FIG. 4. Observed and simulated consumption (in log).

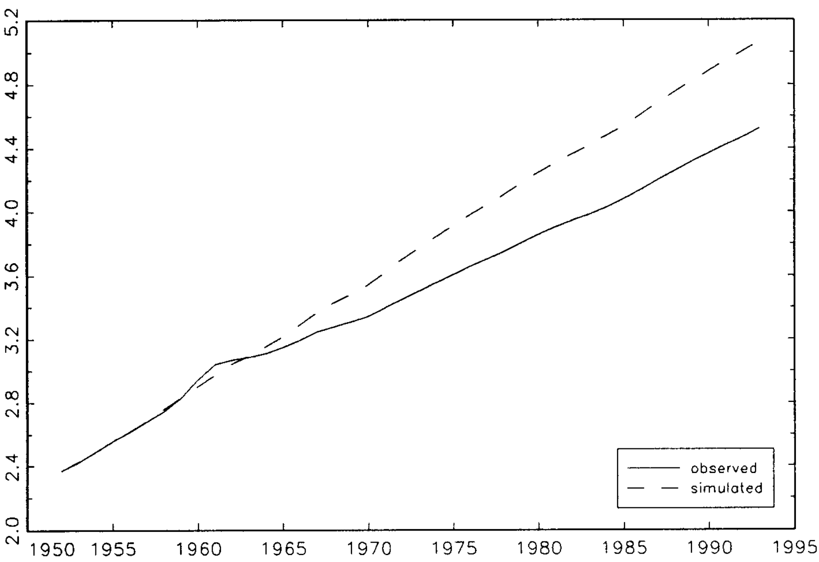


FIG. 5. Observed and simulated capital (in log).

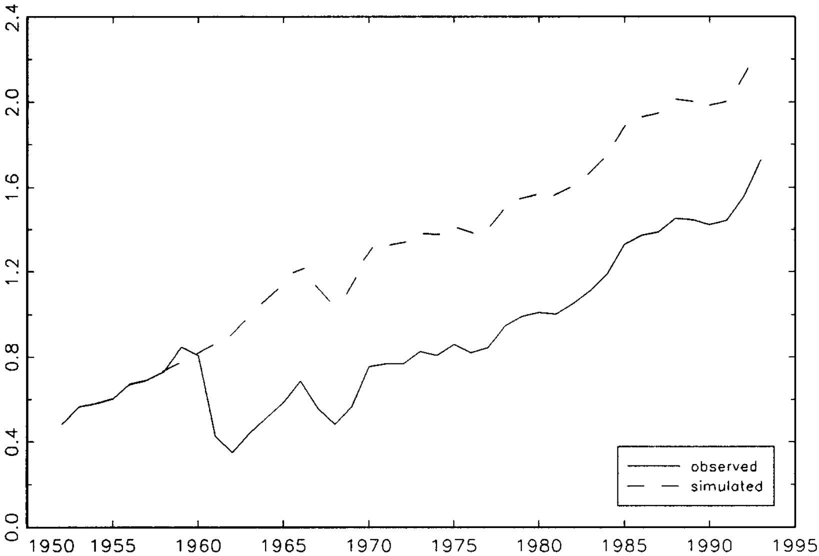


FIG. 6. Observed and simulated Solow residual (in log).

To see the extent of the permanent level shift, we generated 500 residuals of zero mean and covariance matrix given in (15) and appended them to the observed residuals as well as our modified residuals. Output, consumption, and capital were calculated according to these two extended residuals series. Examining the last 100 entries revealed that the steady state had been attained, as evident by the balanced growth of the three variables. Taking the ratio of the two output series gives the permanent level effect which we report in Table 3 in the row labeled as steady state.

To assess the effect of the Cultural Revolution and the combined effect of the two movements, we have performed a similar simulation exercise as that

TABLE 3

SIMULATION/OBSERVED LEVEL IN 1992

	Great leap	Cultural revolution	Both
Output	2.0031	1.2033	2.7130
Consumption	2.0047	1.2022	2.7261
Capital	1.7208	1.1537	2.1687
Steady state	2.1074	1.2204	2.9238

Note. $(\hat{\alpha}, \hat{\beta}, \hat{\gamma}) = (0.7495, 0.9999, 0.0218)$.

TABLE 4

SIMULATION/OBSERVED LEVEL IN 1992

	Great leap	Cultural revolution	Both
Output	2.5446	1.2355	3.6549
Consumption	2.5680	1.2349	3.7277
Capital	1.9708	1.1643	2.5461
Steady state	3.2856	1.3111	5.2465

Note. α fixed at 0.5, $\hat{\beta} = 0.9715$, $\hat{\gamma} = 0.0083$.

described above by removing residuals of the turbulent years. Table 3 provides a short summary for comparison with the Great Leap case; tables similar to Table 2 are available on request. For example, the output level by 1992 would have been 2.7 times higher than otherwise if both political movements had never occurred. To show the degree of sensitivity of our results, Tables 4 and 5 give similar comparisons when other parameter values reported in Section 3 are used.

Absent the Cultural Revolution, output in China in 1992 would have been 1.20 times as large as the actual figure. This estimate might be considered too small. The possibility of under-estimation is mainly due to the omission of the effect on human capital formation in our model. Given that human capital is not considered within the confines of our model, the measured effect appears reasonable. The disruption of the Cultural Revolution in the production of physical output in China is recognized to be much smaller than the disruption of the Great Leap. The relative magnitudes of 1.2 and 2.0 seem quite plausible. The Cultural Revolution is known for its effect on the production of human capital when many schools and universities were closed or ceased to function properly. The estimate of 1.2 can serve as a benchmark for studying the effects of the Cultural Revolution through its effect on the accumulation of human capital.

5. CONCLUSIONS

We have constructed a very simple econometric model to measure the effects of two major political events in China. The model is based on a dynamic optimization framework. It is assumed that an economic planner in China tries to maximize a multiperiod object function in making consumption and investment decisions. The values of the parameters of the optimization model as estimated by maximum likelihood are reasonable. The dynamic optimization framework is useful for studying economic behavior and the effects of political events in China as in other countries.

TABLE 5

SIMULATION/OBSERVED LEVEL IN 1992

	Great leap	Cultural revolution	Both
Output	2.2907	1.2217	3.2082
Consumption	2.3008	1.2207	3.2459
Capital	1.8614	1.1597	2.3796
Steady state	2.6306	1.2648	3.9152

Note. α fixed at 0.6, $\hat{\beta} = 0.9817$, $\hat{\gamma} = 0.0132$.

Concerning the effects of the Great Leap and the Cultural Revolution, our results indicate that absent the former output and consumption per laborer in 1990 would have been 2.0 times as large as the observed, that absent the latter output and consumption would have been 1.2 times as large, and that if neither had occurred output and consumption would have been 2.7 times the actual amounts.

APPENDIX

A standard dynamic optimization problem is to choose a sequence of q by 1 control vectors $\{u_t, t = 0, 1, 2, \ldots\}$ to solve

$$\max E_0 \sum_{t=0}^{\infty} \beta^t r(x_t, u_t) \tag{A1}$$

subject to

$$x_{t+1} = f(x_t, u_t) + \epsilon_{t+1}, \tag{A2}$$

where E_0 is the conditional expectation operator given information at time 0, x_t is a p by 1 vector of state variables, and ϵ_t is an iid random vector with mean zero and covariance matrix Σ. Our problem is to solve

$$\max E_0 \sum_{t=0}^{\infty} \beta^t \ln\{\overline{k}_t^{1-\alpha} \overline{z}_t^{\alpha-1} - \overline{k}_{t+1} + \overline{k}_t \overline{z}_t^{-1}\} \tag{A3}$$

subject to

$$\ln \overline{z}_{t+1} = \mu + \epsilon_{t+1}. \tag{A4}$$

Our problem can be mapped into the standard form by defining the states and control as

$$x_t \equiv (x_{1t}, x_{2t})' = (\ln \bar{z}_t, \ln \bar{k}_t)', \qquad u_t = \ln \bar{k}_{t+1}. \tag{A5}$$

The objective function and the constraint are, respectively,

$$r(x_t, u_t) = \ln\{\exp((1 - \alpha)(x_{2t} - x_{1t})) - \exp(u_t) + \exp(x_{2t} - x_{1t})\} \tag{A6}$$

and

$$f(x_t, u_t) = Ax_t + Cu_t + b, \tag{A7}$$

where A is a 2 by 2 zero matrix, $C = (0, 1)'$, and $b = (\mu, 0)'$. The steady state $(\bar{u}, \bar{x})$ can be found by solving a deterministic, time invariant version of the first order conditions. For our choice of state and control as in (A5), the steady state values are

$$\bar{u} = -\frac{1}{\alpha}\ln[\beta^{-1}\exp(\mu) - 1] + \frac{1}{\alpha}\ln(1 - \alpha) + \mu, \quad \bar{x}_1 = \mu, \quad \bar{x}_2 = \bar{u}. \tag{A8}$$

Only in exceptional cases would one be able to find an analytical solution for the optimal control function. In most applications one has to rely on numerical approximation. One convenient way to do so has been developed in Chow (1992, 1993a). We now describe briefly the solution procedure. Consider the first order conditions

$$r_2(x_t, u_t) + \beta f_2'(x_t, u_t)E_t\lambda_{t+1} = 0, \tag{A9}$$

$$r_1(x_t, u_t) + \beta f_1'(x_t, u_t)E_t\lambda_{t+1} = \lambda_t, \tag{A10}$$

$$x_{t+1} = f(x_t, u_t) + \epsilon_{t+1}, \tag{A11}$$

where the subscripts 1 and 2 of the functions r and f denote derivatives with respect to the first and second arguments, respectively. λ is a vector of random Lagrange multipliers. We proceed by linearizing the non-linear functions in (A9)–(A11) around the steady state $(\bar{x}, \bar{u})$:

$$f = Ax + Cu + b; \qquad r_1 = K_{11}x + K_{12}u + k_1;$$

$$r_2 = K_{21}x + K_{22}u + k_2. \tag{A12}$$

Given the linear functions above, if λ is assumed to be linear, say equal to $Hx + h$, substituting these functions in the first order conditions will yield a linear control function

$$u = Gx + g, \tag{A13}$$

where

$$G = -(K_{22} + \beta C'HC)^{-1}(K_{21} + \beta C'HA) \tag{A14}$$

$$g = -(K_{22} + \beta C'HC)^{-1}(k_2 + \beta C'(Hb + h)), \tag{A15}$$

and the coefficient matrices of the Lagrangean function are, respectively,

$$H = K_{11} + K_{12}G + \beta A'H(A + CG) \tag{A16}$$

$$h = (K_{12} + \beta A'HC)g + k_1 + \beta A'(Hb + h). \tag{A17}$$

Iterating the matrix equation system (A14)–(A17) until convergence gives the required matrices G, g, H, and h. We have accelerated such a direct iteration scheme by incorporating a modified version of the doubling algorithm described in Anderson and Moore (1979, p. 159). A detailed discussion of the algorithm and numerical examples will be reported elsewhere.

REFERENCES

Anderson, Brian D., and Moore, John B., *Optimal Filtering*. New Jersey: Prentice-Hall, 1979.

Chow, Gregory C., *Econometrics*. New York: McGraw-Hill, 1983.

Chow, Gregory C., ''Dynamic Optimization Without Dynamic Programming.'' *Econom. Modelling* **9:**3–9, January 1992.

Chow, Gregory, C., ''Optimal Control Without Solving the Bellman Equation.'' *J. Econom. Dynam. Control* **17:**621–630, July 1993a.

Chow, Gregory, C., ''Capital Formation and Economic Growth in China.'' *Quart. J. Econom.* **108:** 809–842, August 1993b.

Goffe, William L., Ferrier, Gary, and Rogers, John, ''Global Optimization of Statistical Function.'' In Hans M. Amman, David A. Belsley and Louis F. Pau, Eds., *Computational Economics and Econometrics,* Vol. 1. Dordrecht: Kluwer, 1992.

Li, Jingwen, Gong, Feihong, and Zheng, Yisheng, ''Productivity and China's Economic Growth, 1953–1990.'' In Kai-yuen Tsui, Tien-Tung Hsueh and Thomas G. Rawski, Eds., *Productivity, Efficiency and Reform in China's Economy*. Hong Kong: Chinese Univ. Press, 1995.

Lin, Justin Y., ''Collectivization and China's Agricultural Crisis in 1959–1961.'' *J. Polit. Econom.* **98,** 6:1228–1252, December 1990.

Press, William H., Teukolsky, Saul A., Vetterling, William T., and Flannery, Brian P., *Numerical Recipes in C: The Art of Scientific Computing,* Second Edition. New York: Cambridge Univ. Press, 1992.

Statistical Yearbook of China. Beijing: State Statistical Bureau, 1994.