

Progetto **Web Math**

prof. Roberto Fuligni

Istituto Tecnico Tecnologico “Giacomo Fauser” - Novara

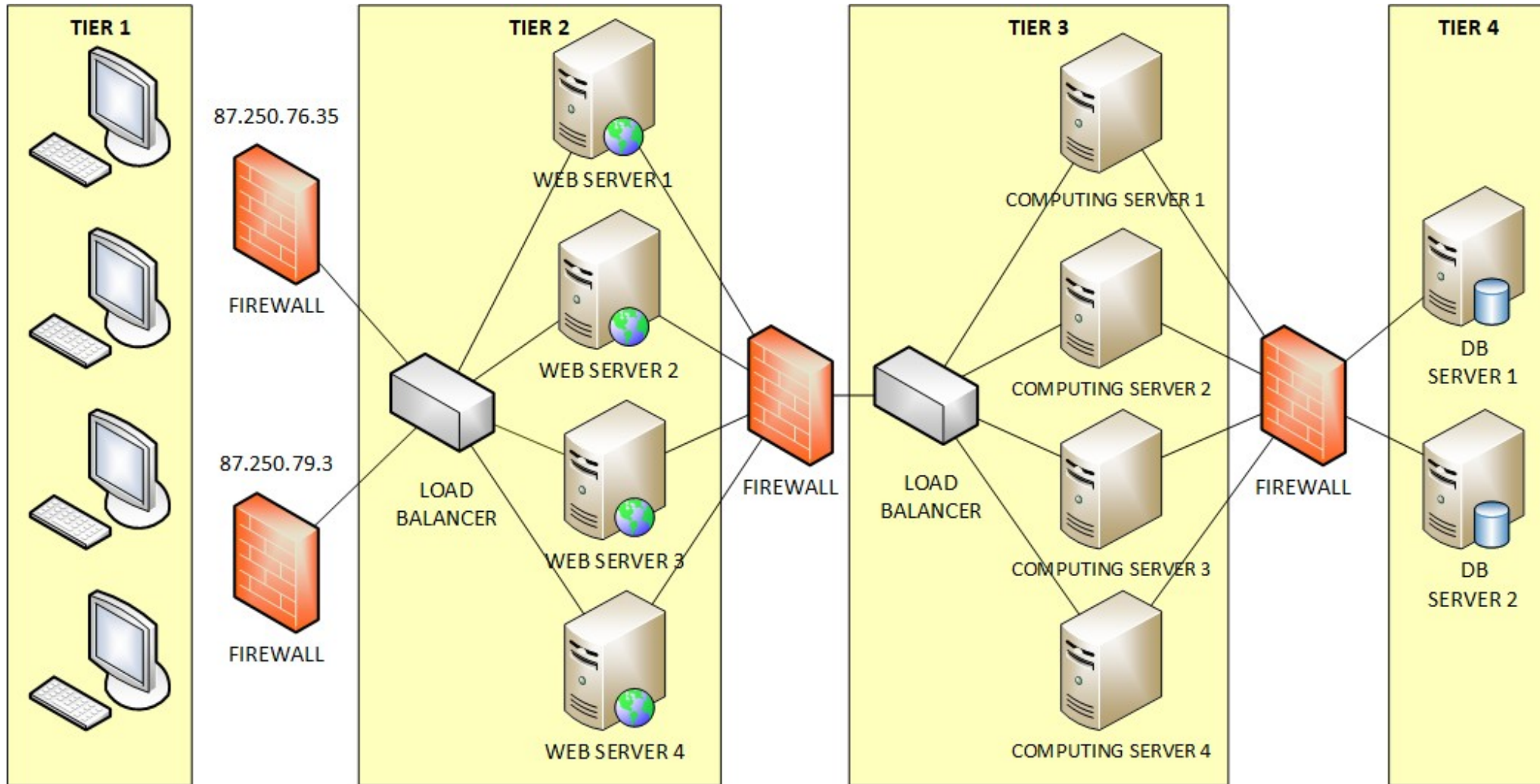
Web Math

- Applicazione di rete per la rappresentazione grafica di funzioni matematiche
- Architettura client/server
- Progetto realizzato con tecnologie Microsoft
 - Software sviluppato in C#
 - Comunicazione tra client e server basata su HTTP e WCF (*Windows Communication Foundation*)
 - Supporto di client eterogenei mediante Api RESTful

Web Math

- **Modulo server installato sulla piattaforma Netlab**
 - Servizi utilizzabili anche all'esterno dell'istituto
 - Elevata disponibilità di servizio grazie all'impiego di apparecchiature ridondate e bilanciamento del carico
- **Architettura Four-Tier**
 - 1) *Client* con interfaccia WPF (*Windows Presentation Foundation*)
 - 2) *Web server* ospitato su IIS (*Internet Information Services*)
 - 3) *Application server* per il rendering dei grafici
 - 4) *Database server* per la memorizzazione dei dati di utilizzo

Schema architetturale



Funzionamento del client

- Il client contatta il servizio mediante l'URL **<http://services.netlab.fausser.edu>**
 - Il nome simbolico è associato agli indirizzi pubblici di due firewall (87.250.76.35 e 87.250.79.3)
 - Il client risolve il nome con DNS e sceglie uno dei due indirizzi (round robin)
- Invia una richiesta contenente la funzione da rappresentare e altre informazioni

Funzionamento dei server

- Le richieste dei client sono consegnate a un *load balancer* che smista i messaggi ai server web collegati
 - La suddivisione del carico di lavoro migliora l'efficienza complessiva in presenza di numerose richieste
- Il web server verifica la correttezza della richiesta
 - Sintassi della funzione, validità dei parametri di rappresentazione
 - In caso di errore genera un'eccezione che sarà segnalata al client

Funzionamento dei server

- Se la richiesta è valida, il web server apre una connessione TCP verso un server interno (*computing server*)
 - Trasmette la funzione matematica da rappresentare
- Il *computing server* elabora l'immagine (rendering) e restituisce il risultato sulla connessione TCP
- Il web server incapsula il grafico di funzione in un apposito messaggio di risposta
 - La risposta è successivamente consegnata al client richiedente

Implementazione del client

- La comunicazione con il server si avvale di due *Data Contract WCF*
 - Classe **PlotRequest**: specifica i dettagli della richiesta
 - Funzione da rappresentare (è possibile richiede i grafici sovrapposti di due funzioni)
 - Sono supportate sia le funzioni in una variabile ($f(x)$, grafico 2D) sia quelle in due variabili ($f(x,y)$, grafico 3D)
 - Intervallo di rappresentazione (*range*)
 - Dimensioni dell'immagine in pixel

Implementazione del client

- Classe **PlotResponse**: include il grafico di funzione e altre informazioni sul processo di elaborazione
 - Array di byte contenente l'immagine in formato binario
 - Dimensioni dell'immagine al termine dell'elaborazione (possono differire da quelle richieste)
 - Contatore di elaborazioni effettuate dal servizio
 - Nomi dei server che hanno processato la richiesta
- **La richiesta è inviata attraverso un apposito *Operation Contract WCF***
 - *GetPlot*

Struttura di PlotRequest

```
[DataContract]
public class PlotRequest
{
    [DataMember]
    public string function { get; set; }

    [DataMember]
    public string function2 { get; set; }

    [DataMember]
    public string XRange { get; set; }

    [DataMember]
    public string YRange { get; set; }

    [DataMember]
    public int Height { get; set; }

    [DataMember]
    public int Width { get; set; }
}
```

Struttura di PlotResponse

```
[DataContract]
public class PlotResponse
{
    [DataMember]
    public byte[] ImageData { get; set; }

    [DataMember]
    public int Height { get; set; }

    [DataMember]
    public int Width { get; set; }

    [DataMember]
    public int PlotCounter { get; set; }

    [DataMember]
    public string WebServer { get; set; }

    [DataMember]
    public string ComputingServer { get; set; }
}
```

Struttura del contratto di servizio

```
[ServiceContract]
public interface IWebMath
{
    [OperationContract]
    PlotResponse GetPlot(PlotRequest request);
}
```

Generazione delle classi

- *Web Math* supporta la generazione automatica del client WCF e dei contratti mediante lo scambio di metadati
 - In questo modo non è necessario definire manualmente le classi e le interfacce dei contratti
- **Per importare le classi in un progetto Visual Studio**
 - Fare click sul progetto con il tasto destro del mouse , quindi selezionare **Aggiungi** → **Riferimento al servizio**
 - Specificare l'indirizzo
<http://services.netlab.fausser.edu/webmath/WebMath.svc>

Esempi d'uso

- Richiesta del grafico di $f(x)=\cos(2*x)+\sin(3*x)$ nell'intervallo $[-5, 5]$ (immagine 300x200 pixel)

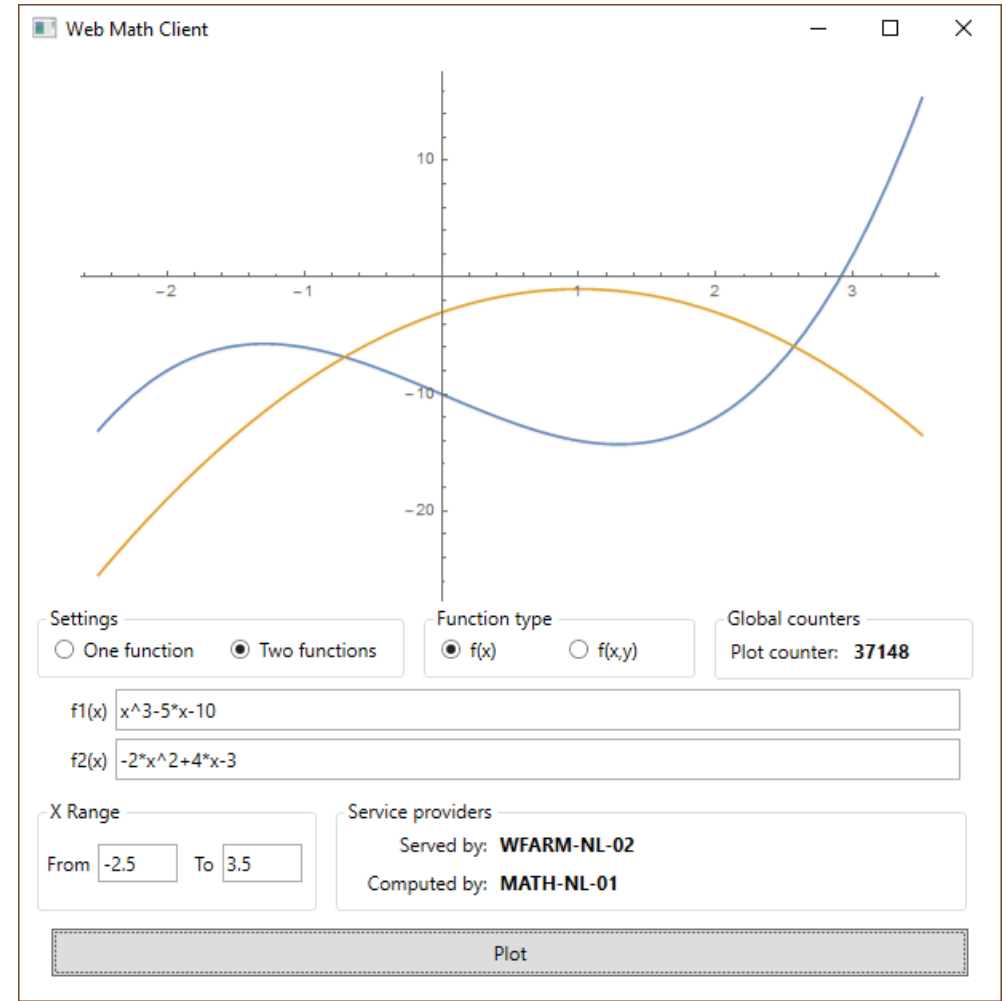
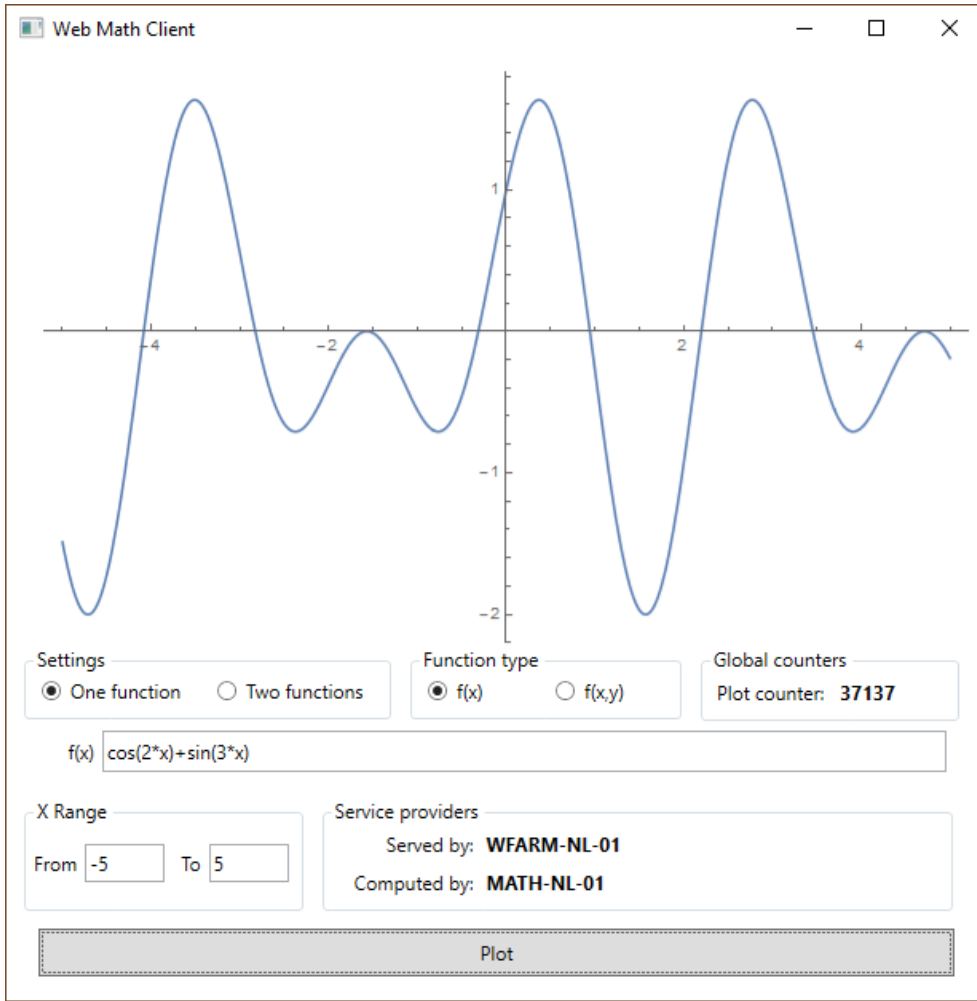
```
ServiceReference1.PlotRequest req;  
ServiceReference1.PlotResponse resp;  
req = new ServiceReference1.PlotRequest();  
req.Width = 300;  
req.Height = 200;  
req.function = "cos(2*x)+sin(3*x)";  
req.XRange = "-5; 5";  
var sc = new ServiceReference1.WebMathClient();  
resp = sc.GetPlot(req);
```

Esempi d'uso

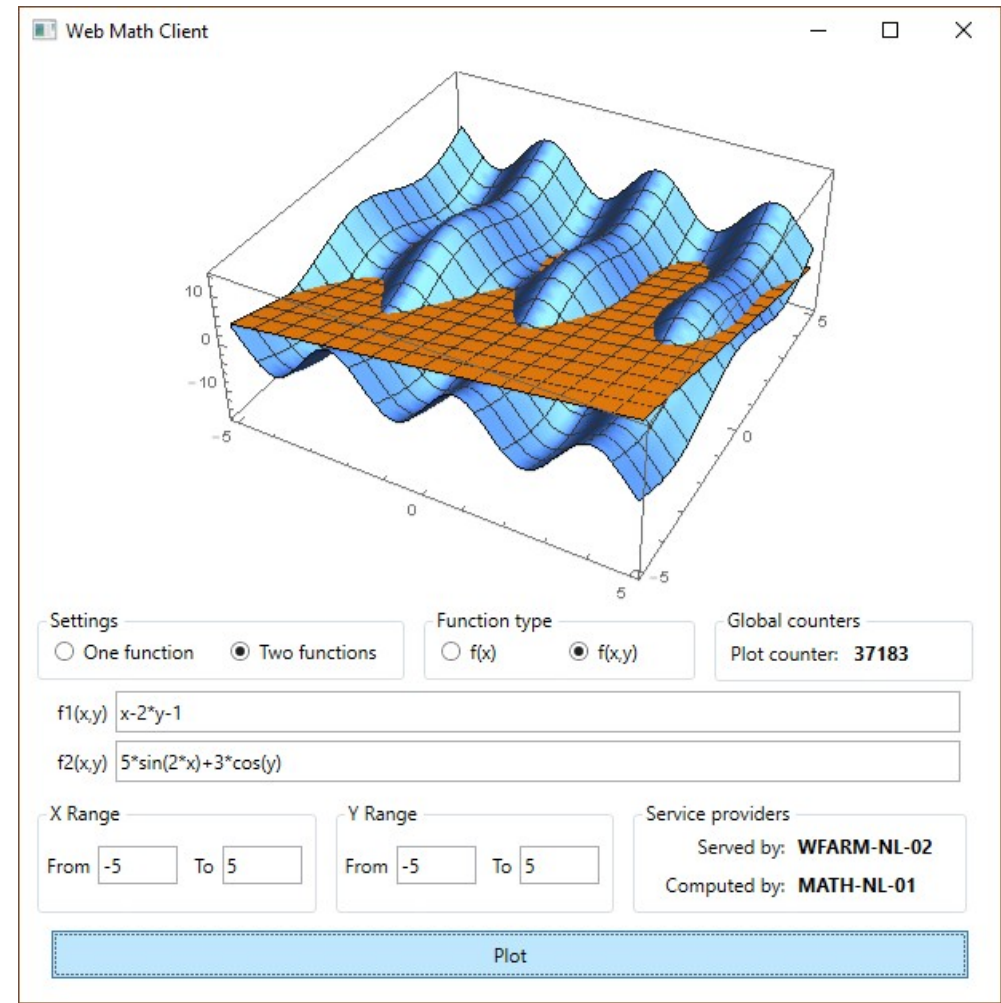
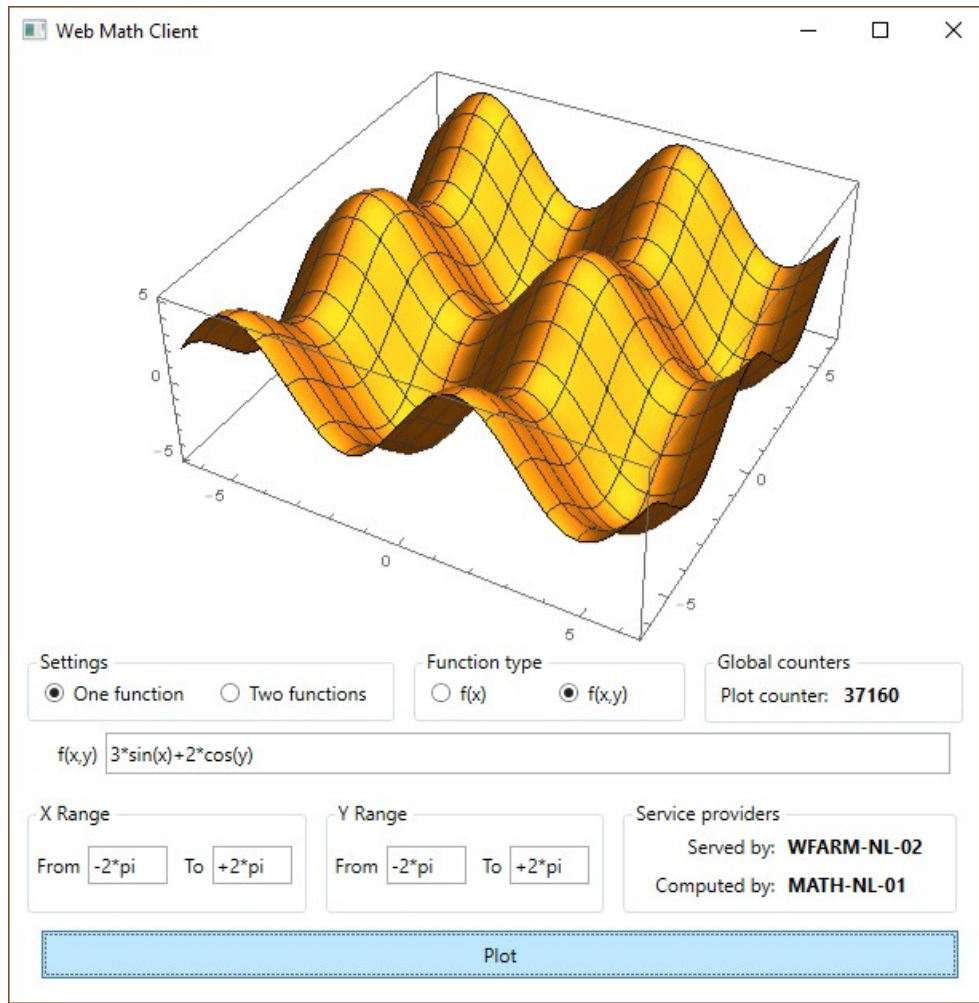
- **Visualizzazione del grafico in un componente WPF di tipo “Image”**

```
BitmapImage biImg = new BitmapImage();  
var ms = new MemoryStream(resp.ImageData);  
biImg.BeginInit();  
biImg.StreamSource = ms;  
biImg.EndInit();  
// "Immagine" e' il nome di un componente WPF  
Immagine.Source = biImg as ImageSource;
```

Esempio di client GUI (grafico 2D)



Esempio di client GUI (grafico 3D)



Web Api

- Il web server offre un insieme di Web Api per la costruzione di grafici
 - Possibilità di utilizzare tecnologie diverse nello sviluppo del client (es. web client)
 - <https://services.netlab.fausser.edu/webmath/api/plot>
 - Metodo POST
 - Client e server scambiano oggetti in formato JSON
 - Struttura identica a PlotRequest e PlotResponse
 - La risposta contiene l'immagine elaborata in formato Base64
 - Gli errori sono segnalati mediante appositi codici di stato HTTP

Frammento di web client (Javascript)

```
<p><img id="imgresult" /></p>
<button onclick="updateimg();">Update</button>

<script>
  function updateimg() {
    let request = {
      "function": "sin(x)",
      "XRange": "-10;10",
      "Height": "200",
      "Width": "300"
    }

    /* Acquisizione del grafico mediante Fetch API Javascript */

    fetch('https://services.netlab.fausser.edu/webmath/api/plot', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(request),      // Trasforma la richiesta in una stringa JSON
    })
    .then(response => {
      // Fase 1: Il body della risposta è estratto dalla risposta in formato JSON
      // ed è passato alla fase successiva (modalità asimmetrica)
      return response.json()
    })
  }
}
```

Frammento di web client (Javascript)

```
.then(data => {
  // Fase 2: Elaborazione dei dati ricevuti

  // L'eventuale presenza del campo "Message" indica un errore di elaborazione
  if (data.Message)
    throw new Error(data.Message)

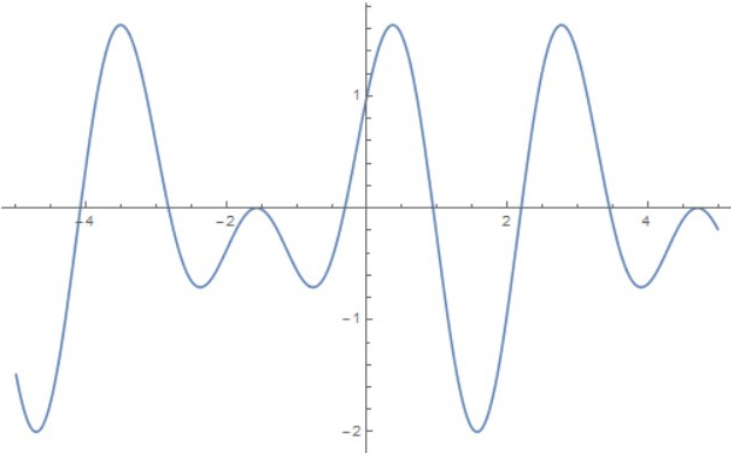
  // Visualizzazione su console di tutti i dettagli della risposta
  console.log(data)

  let img = document.getElementById("imgresult")

  // Si associa all'elemento <img /> un'immagine PNG codificata in formato Base64
  img.setAttribute("src", "data:image/png;base64, " + data.ImageData)
})
.catch((error) => {
  // Tutti gli errori sono riportati sulla console del browser
  console.error(error)
})
}
</script>
```

Web client (grafico 2D)

Web Math



Served by: **WFARM-NL-03** Computed by: **MATH-NL-02** Plot counter: **45808**

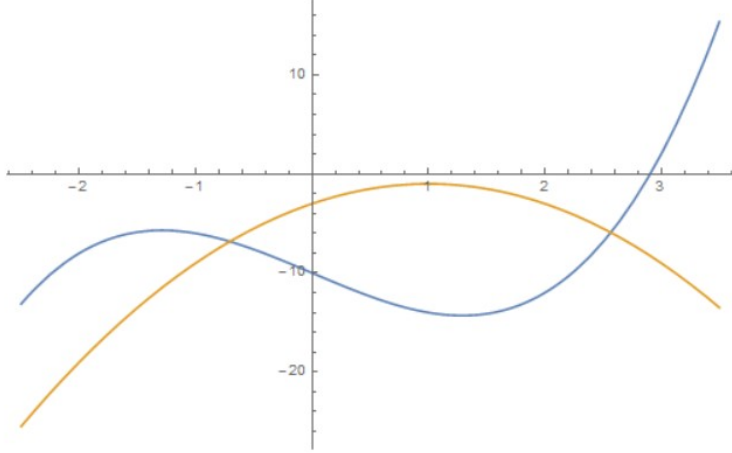
Settings: One function Function type: f(x)

f1(x): $\cos(2*x) + \sin(3*x)$

X Range (From: -5 X Range (To: 5

Plot

Web Math



Served by: **WFARM-NL-01** Computed by: **MATH-NL-04** Plot counter: **45809**

Settings: Two functions Function type: f(x)

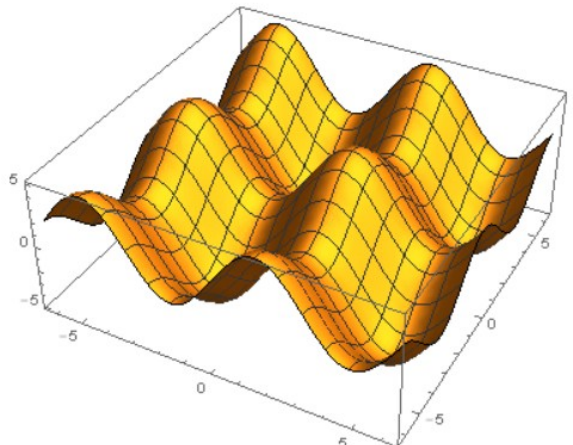
f1(x): $x^3 - 5*x - 10$ f2(x): $-2*x^2 + 4*x - 3$

X Range (From: -2.5 X Range (To: 3.5

Plot

Web client (grafico 3D)

Web Math



Served by: **WFARM-NL-03** Computed by: **MATH-NL-03** Plot counter: **45810**

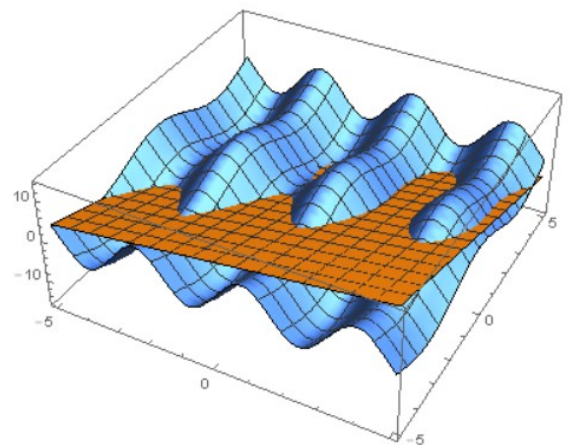
Settings: One function Function type: $f(x,y)$

f1(x): $3*\sin(x)+2*\cos(y)$

X Range (From:): $-2*\pi$ X Range (To:): $2*\pi$ Y Range (From:): $-2*\pi$ Y Range (To:): $2*\pi$

Plot

Web Math



Served by: **WFARM-NL-01** Computed by: **MATH-NL-01** Plot counter: **45811**

Settings: Two functions Function type: $f(x,y)$

f1(x): $x-2*y-1$ f2(x): $5*\sin(2*x)+3*\cos(y)$

X Range (From:): -5 X Range (To:): 5 Y Range (From:): -5 Y Range (To:): 5

Plot